UNIVERSITÉ
DE GENÈVE
FACULTY OF ECONOMIC
AND SOCIAL SCIENCES
Department of Information Systems

Performance Measurements

Designing a Generic Measure and Performance Indicator Model

Master Thesis submitted by

Othmar Heini

for the degree of

Master of Science in Communication and Information Systems
Orientation Information System Technologies

Under the supervision of Prof. D. Konstantas, M. Pawlak and K. Wac

Geneva, 2007

Acknowledgement

First of all, I wish to thank Professor Dimitri Konstantas and Professor Michel Léonard for their precious guidance and advice over the years that I have studied at the Department of Information Systems of the University of Geneva.
Special thanks are due to Michel Pawlak, who kept me going in this research, and to Katarzyna Wac for her valuable comments and advice. You are a true researcher! I would like to thank all my fellow students, especially Wanda and Guillaume, as well as all the members of the ASG and MATIS team for their joyful company.
I would also like to thank my family and all my friends who have both encouraged and supported me throughout this project.
Finally, I wish to thank Fabienne, for her patience and understanding.

Abstract

Performance measurement has become a major issue in recent years. The ability of an organisation to measure its vital activities, and this at all organisational levels, has indeed become critical to its success in today's fast-paced world. However, current knowledge in the field of performance measurement is still insufficiently translated into practice. For instance, performance indicators are rarely linked to the objectives and the overall vision of the organisation. Time-critical activities are often not supported by current measurement systems, and data from external sources is not sufficiently taken into consideration. Such issues are to some extend due to the absence of a comprehensive, overall model. Only few models have been proposed in literature, most of which lack detail, or do not consider all relevant aspects.

In this research, we present a generic measure and performance indicator model that incorporates the essential aspects of the performance measurement process in a single, comprehensive model. Based on an extensive literature review in the fields of measurement and performance measurement, we identify key concepts and discuss major theories and frameworks. From our findings, we derive a set of requirements, which we translate into our model. Finally, we propose a generic architecture for performance measurement systems, and present a prototype application which builds upon our model proposal.

Contents

List of Figures

List of Tables

Chapter 1.

Introduction

1.1. Introduction to Measurement and Performance Measurement

Measurements are processes that we accomplish on a regular basis, and in a variety of domains and circumstances. When leaving for work for example, we may glance at the thermometer in order to decide whether or not we should take our coat; a craftsman constantly controls the size of the piece he is working on to make sure it fits together with the others; a sales manager periodically checks the sales-volume of a product or service in order to compare it with forecasts.

The central idea in *measurement* is the notion of representation, and more precisely 'the translation of "qualitative" concepts such as relations and operations into appropriate "quantitative" [. . .] relations and operations enjoying known properties' [105]. In measuring weight for example, we seek to assign numbers to the weight property of objects in a way that relations such as 'heavier than' and 'lighter than', and operations such as addition or subtraction remain preserved. Measurement of objects and phenomena is an important aspect in all sciences, for it is through observation, generalisation and measurement that theories can be derived. For instance, where would the physical sciences stand today, if it had not been able to define measurement standards for properties such as weight and length?

The measurement of organisational performance has become a major issue in recent years. In this context, *performance* can be considered as 'a task or operation seen in terms of how successfully it is performed' [95]. The measurement of performance, referred to as *performance measurement*, is a process which allows an organisation to evaluate whether it is on track, or whether actions should be initiated that help achieving its objectives. Early performance measurement focused mainly on financial measures. This uni-dimensional approach has been criticised in the 1980's by various authors. Changes in the world markets and the increasing competition that enterprises were facing in the 1990's led to new approaches to performance measurement. Besides a more 'balanced', multi-dimensional approach, emphasis is put on strategy and objectives. The *performance indicator* is the main instrument of the performance measurement process. We define it as a strategic instrument which allows to evaluate performance against tar-

gets.
The importance of measuring performance on a regular basis is widely accepted today (surveys can be found in [29, 31, 81, 116]). Besides traditional financial measures such as profit and return on investment, organisations also evaluate non-financial measures in fields such as operations and customer satisfaction. Measurements generally support the activities of different departments, and are used at different organisational levels. While performance measurement is critical to profit-oriented organisations, it is also becoming increasingly important for non-profit and public organisations. For instance, performance measurement is present in fields such as public management [59, 85], healthcare [67, 82, 86], and education [53, 113]. Non-profit and public organisations differ from their counterparts in the private sector in that they are not focused on profit maximisation and generate income in a different way. Factors that motivate the trend towards performance measurement in such organisations include a more market-oriented view of state, constricted resources, the pursuit of efficiency, and demands for accountability [53].
An area that has seen recently an increased interest is the measurement of processes. Indeed, the success (or failure) of an organisation depends to a large degree on the quality of its processes, which is why a lot of effort has been directed in the last decade to the reengineering of processes. The measurement of process performance enables the concerned actors to assess where they stand and verify that the required performance is delivered. Furthermore, it allows to recognise problems and detect opportunities for process improvement. Not surprisingly, measurements play an important role in current standards and models for process maturity and improvement, notably in the Capability Maturity Model Integration (CMMI) [109], and the ISO/IEC 15504 and ISO 9000 standards.

1.2. Motivation

The importance of measuring performance on a regular basis is widely accepted and acknowledged by academics and practitioners. Research conducted in the last two decades led to numerous recommendations and frameworks, which provide as a solid, theoretical foundation for the measurement activities in organisations.
The importance of automating some, if not all aspects of performance measurement—typically the collection, processing and distribution of performance information—has been identified early. Globerson [49], for instance, recommended in 1985 to collect performance data as part of work processes. He further notes that data gathering may be part of the computer program supporting the processes. Other authors, such as Bourne et al. [19] and Bititci et al. [15], further highlight the importance of information systems in the performance measurement process. Today, various software editors propose commercial solutions to performance measurement, and a few systems have been proposed by researchers.
However, current knowledge in the field of performance measurement is still insufficiently translated into practice, notably in the following areas:

- links to objectives and initiatives,
- timeliness of performance information,
- consideration of external data sources,
- relationships between performance indicators,
- reusability of measures,
- standardisation.

We briefly discuss each of these aspects.

Links to objectives and initiatives One of the most often stated recommendation in performance measurement is to derive performance indicators from strategy and objectives [49, 51, 62, 74, 79, 88]. Once the overall strategy of the organisation is clarified, it needs to be translated into a set of objectives for which appropriate performance indicators are to be identified. Deriving performance indicators from strategy and objectives ensures that the whole organisation is pulling in the same direction, avoiding thereby local sub-optimisations.
However, current performance measurement systems rarely explicit the link between objectives and performance indicators, instead, they focus on the indicators and their targets. Similarly, initiatives that are intended to support the objectives and improve performance are rarely linked to objectives and performance indicators. Kaplan and Norton [63] note that although many organisations have initiatives under way, they are frequently not linked to achieving targeted improvement for objectives.
A performance measurement system should therefore allow performance indicators to be linked to objectives and initiatives.

Timeliness of performance information A popular approach to performance measurement is to extract measurement data from a centralised, integrated data source [70, 75, 116]. Data is extracted periodically from operational data sources, transformed, aggregated and finally stored in a central data source, typically in a database or data warehouse. This approach is suitable for measurements that support strategic and tactical purposes, since real-time data is not required. However, the latency introduced by the periodic collection of data may be unsuitable in other areas.
This brings us to the concept of *timeliness*. Forza and Salvador [41] note that information timeliness is a relative concept, which depends upon the time span of the decisions that have to be taken. They argue that the speed of performance feedback has to be higher at lower hierarchical levels, where feedback is supposed to support those continuous adjustments often needed. Golfarelli et al. [50] argue that, in addition to the traditional data warehouse approach, a reactive component capable of monitoring the

time-critical operational processes is required. Similarly, Kueng [69] notes that enterprise applications are rarely adequate for process performance measurement. He argues that process performance measurement systems should be 'conceptualized as a modular, separate information system which is loosely coupled to other information systems throughout the organization'.
Therefore, a performance measurement system should not only be able to source data from integrated, but also from operational data sources in order to be able to support time-critical processes.

Consideration of external data sources Traditionally, performance measurement is based on data that is produced and collected within the boundaries of an organisation, adopting thereby an inward-looking, self-centred attitude. Bititci et al. [15] note, however, that performance measurement systems should not only be sensitive to changes in the internal, but also the external environment of the organisation. Partner organisations and third party information providers for instance may provide the organisation with valuable information on changes and opportunities in its environment.
Access to external information is even more important in the context of supply chains[1], extended[2] and virtual organisations[3]. Each of these concepts describes a different level of strategic collaboration between organisations, the supply chain representing the lowest level, and the virtual organisation the highest level of collaboration. Access to and exchange of information is viewed by many authors as an absolutely necessary and indisputable component in any successful enterprise network [23, 55, 115]. Performance information needs to be accessible throughout the network of organisations to allow the identification of bottlenecks and unexpected changes. Inter-organisation measures further allow the various partners to coordinate their efforts effectively and efficiently.
Integrating external data into the central data store(s) of an organisation may not always be possible, since the organisation does not own the data and may not be allowed to store it locally. Thus, a performance measurement system should not only be able to source data from internal, but also external data sources.

Relationships between performance indicators Performance indicators are rarely independent from one another. Often, they are related and affect each other. Typically, two performance indicators may tend to change in a similar way, or a change in one indicator may cause a change in another.
Bititci et al. [15] note that a performance measurement system should facilitate the quantification and management of the causal relationships between performance indicators. Similarly, Kaplan and Norton [63] note that 'the measurement system should make the

[1] A supply chain is 'a number of organizations—at least three—working cooperatively with at least some shared objectives' [55].

[2] An extended organisation is 'a knowledge-based organisation which uses the distributed capabilities, competencies and intellectual strengths of its members to gain competitive advantage to maximise the performance of the overall extended enterprise' [14].

[3] A virtual organisation is 'a set of companies that acts integratedly and organically; it is constantly re-configured to manage each business opportunity a customer presents' [115].

relationships among objectives (and measures) in the various perspectives explicit so that they can be managed and validated'. Knowledge about the relationships between indicators is valuable, since it provides a better understanding of the dynamics of the particular domain and allows the management of possible conflicts.
Several approaches to the modelling and quantification of the relationships between performance indicators have been proposed in literature [8, 100, 112]. However, few systems actually provide such functionalities. Therefore, a performance measurement system should provide means to quantify and manage the relationships between performance indicators.

Reusability of measures Measurement theory distinguishes between two types of measurement: fundamental measurement and derived measurement [105, 111]. Fundamental measurement is the process at which fundamental measures are defined, by mapping an observed or empirical system to a numerical system. Derived measurement is the process at which new measures are defined in terms of existing ones. Thus, measures should be considered as simple data providers that can be combined through arithmetic and/or logical operators to allow the definition of new, derived measures. Furthermore, measures should be parametrisable in order to promote reusability of the existing.
However, few systems actually support reusability of measures, either because the formulas are hardcoded in the system, or because the system does nothing more than reproduce data stored in a (integrated) data source. Therefore, measures should be considered as parametrisable data providers that can be combined in order to create new measures.

Standardisation Throughout an organisation, numerous initiatives involving measurements usually do exist, often in different departments and at different organisational levels. However, integrating the different measures and performance indicators as to gain an overall view may turn out to be difficult. Most likely, they are not based on a standardised definition, or may even be documented in rather ambiguous ways.
The findings of a case study conducted by Lohman et al. [76] on integrated performance measurement in supply chain management point to the central role of a shared set of standardised performance indicators, which the authors call the metric dictionary. This dictionary holds the definitions of all existing and proposed performance indicators based on a standardised template. Other authors, such as Eckerson [30], also recommend to define measures and performance indicators based on standardised definition, so they can be integrated easily across the organisation.
However, only few models of measures and performance indicators have been proposed in literature. While some of them only briefly describe the most relevant concepts involved, others do not consider all aspects important in the field of performance measurement.

1.3. Objective

In the preceding section, we discuss several aspects of performance measurement that are still insufficiently translated into practice. We argue that these issues are to some extend due to the absence of a comprehensive, overall model. Although central to the performance measurement process, few authors actually define what exactly a measure or performance indicator is, what the aspects are that need to be taken into consideration, and how they relate to each other. The few models that have been proposed in literature are not sufficient in our eyes, they either lack detail or do not consider all aspects relevant to the performance measurement process.
The objective of the present research is to propose a generic measure and performance indicator model, that incorporates the essential aspects of the performance measurement process in a single, comprehensive model. A model that supports the various recommendations and frameworks proposed in the measurement and performance measurement literature, and that is sufficiently generic as to be employed in any domain and for any kind of measurement activity.
Based on this objective, we can derive the following research questions:

1. What is a measurement? And what are its theoretical foundations?

2. What is performance measurement? What are the fundamental concepts involved, how are they related, and what approaches to performance measurement do exist?

3. What models do currently exist for measures and performance indicators?

4. What are the requirements on a generic measure and performance indicator model?

5. How could a model representing generic measures and performance indicators look like?

The proposed model is then to be evaluated through a prototype application of a performance measurement system.

1.4. Structure

The remainder of this document is structured as follows.
In chapter 2, we examine different aspects related to measurement. We first review the terminology, and discuss different methods of measurement. The theoretical foundation of measurement, known as measurement theory, is then examined. We study fundamental measurement, derived measurement, and scales, which are the main components of this theory. Finally, quality of measurement and the limits of measurement are discussed.

In chapter 3, we address the field of performance measurement. The terminology and the history of performance measurement is first examined. We discuss the key concepts and review various recommendations and studies that have been proposed in literature. The balanced scorecard framework, one of the most popular approaches to performance measurement, is then presented.
Existing measure and performance indicator models are examined in chapter 4.
The requirements on a measure and performance indicator model are addressed in chapter 5. We first discuss the importance of requirements in the context of information system development, and list the requirements that can be derived from our findings on measurement and performance measurement.
In chapter 6, a measure and performance indicator model is proposed. First, we examine what a model is, the kinds of models that do exist, and the role of models in the context of information system development. We then present our model proposal and describe its elements. We further discuss issues related to the design of the model, and state the patterns that have been used. Finally, a set of rules that apply to the model are listed.
In the chapter 7, we present a prototype application of a performance measurement system that builds upon our model. A generic architecture is first proposed, before we present the prototype application. We describe the use cases the application translates and discuss technical aspects. In a simple scenario we explain the overall functioning, and finally evaluate our model based on the prototype application.
In chapter 8, we present our conclusion and discuss future research.

Chapter 2.

Measurement

The aim of this chapter is to answer our first research question: What is a measurement? And what are its theoretical foundations?
We begin with analysing the terminology related to the measurement domain, and discuss the different methods of measurement. We then examine the measurement theory, a body of research work which acts as a theoretical foundation for measurement. Measurement theory distinguishes between two approaches: fundamental and derived measurement. We study the formal aspects of each of them, as well as the concept of scale, which is central to measurement. We then discuss two notions which are essential to the quality of measurement, and conclude this chapter with a discussion on the limits of measurement.

2.1. The Nature of Measurement

2.1.1. What is a Measurement?

The term *measurement* has a number of distinct meanings, depending on the circumstances in which it is being used. According to the Oxford Dictionary of English [95], the term measurement can be used to designate either 'the action of measuring something', 'the size, length, or amount of something, as established by measuring', or 'a unit or system of measuring'. From a general point of view, the term measurement can thus refer to an action, a result or a standard.
In the context of scientific research, a more specific meaning is usually attributed to this term. Table 2.1 states some of the definitions that can be found in literature. They tend to suggest that measurement is about assigning numbers to the properties of entities in a way that their original characteristics remain preserved. But measurement without numbers is also a perfectly legitimate and useful activity, notes Roberts [105]. For instance, one could assign words, or even symbols to the properties of the observed entities. This may make sense in certain circumstances, but the use of language or symbols restricts considerably the further use and manipulation of the measurements. Pfanzagl [99] notes

Author / Source	Definition
Roberts [105]	The assignment of numbers that correspond to or represent or 'preserve' certain observed relations.
Boyce et al. [21]	The process of acquiring quantitative data about the physical world through the comparison of a known standard unit with a comparable property of an entity that we desire to characterize in a quantitative manner.
Stevens (mentioned in [105])	The assignment of numerals to objects or events according to rules.
Torgerson (mentioned in [105])	Measurement of a property [. . .] involves the assignment of numbers to systems to represent that property.

Table 2.1.: Definitions of the term 'measurement'

on this subject that 'it is of obvious importance to describe the structure of a property in a more accurate and more systematic way than is achieved by language'. Indeed, the assignment of numbers to the observed phenomena makes possible the application of the concepts and theories of mathematics. Thus, scientific analysis can be attained at a finer graduation. As to the reason for measurement, Boyce et al. [21] note that 'measurement allows us to make statements which can be clearly understood and verified by others'. Measurements allow precise descriptions of objects and phenomena and allow observations to be reproduced by others. Moreover, measurements may lead to the derivation of new laws and principles.

Two terms that are frequently used in the context of measurement are *measure* and *metric*. We discuss briefly their meanings.
The Oxford Dictionary of English [95] gives the following two definitions for the verb *measure*: to 'ascertain the size, amount, or degree of (something) by using an instrument or device marked in standard units', and to 'be of (a specified size or degree)'. The noun *measure* represents 'a standard unit used to express the size, amount, or degree of something'. However, in the American English language, this terms is also employed to designate 'a size or quantity found by measuring' [93].
The term *metric* originates from the mid 19th century and derives from the french word *métrique* (from *mètre*). Initially, it was used as an adjective related to length, while nowadays, it is also employed as noun. As to its definition, it seems that it depends to a large degree on the context in which the term is being used. Table 2.2 lists some the definitions that can be found in literature. The definitions vary largely from one author to another. For some authors, a metric is a quantitative measure, for others a function, and for still others a figure or a statistic. However, they all seem to share the same basic idea that a metric is related to a standard of measurement.
In the remainder of this document, we do not use the term metric since its meaning depends too much on the context in which it is being used. Rather, we use the term measure to designate a standard of measurement that allows to represent a property of an entity in a quantitative manner.

Author / Source	**Definition**	**Context**
The Concise Oxford English Dictionary [96]	(Metrics) A set of figures or statistics that measure results.	Business
The Oxford Dictionary of English [95]	A system or standard of measurement.	Technical
Dictionary of Computing [94]	A number representing the degree to which software, or an entity related to software, possesses some notional property, as observed or perceived by an individual or group.	Computing
IEEE [57]	A quantitative measure of the degree to which a system, component, or process possesses a given attribute.	Software Engineering
Purao & Vaishnavi [103]	A function that assigns a number or symbol to an entity in order to characterize an attribute or a group of attributes.	Software Engineering
Gartner, Inc. [46]	A standard of measurement.	Information Technology
Aceituno Canal [25]	A Metric is a quantitative measurement that can be interpreted in the context of a series of previous equivalent measurements.	Information Security

Table 2.2.: Definitions of the term ‘metric’

2.1.2. Absolute and Relative Measurement

All measurement is relative to a chosen unit, this unit can either be absolute or relative. *Absolute measurement* refers according to Boyce et al. [21] to ‘the existence and use of an official standard which has a high degree of acceptance’. In physics for example, the units of measure for basic attributes include meter for length, kilogram for mass, and seconds for time.
Relative measurement on the other hand uses a standard which is meaningful only in a particular context. For example, in the statement ‘John is half as tall as his father’, it is the height of John’s father that is considered as the standard. However, such relative measurement is not meaningful outside the community of people who know approximately how tall John’s father is. ‘Last sunday was cold’ is another example. It expresses a personal evaluation of the temperature, indicating that the temperature was lower than the norms of the speaker. This statement could be true at widely varying quantitative values, depending upon the time of year and other factors, such as humidity and wind speed. On the other hand, another speaker might perceive the same conditions as ‘cool’ or even ‘pleasant’.

2.1.3. Methods of Measurement

Direct Measurement The most obvious method of acquiring quantitative data about the physical world is by *direct measurement*. Numbers can be assigned to objects and phenomena according to some predefined rules. For example, a physical object can be measured with a meter stick. In direct measurement, it is assumed that an observation will be very similar to those made by other observers who use the same standard. However, this approach bears a fundamental problem. Boyce et al. [21] state that ‘direct measurement by observation seems to depend upon the assumption of an external physical world, an assumption which is very difficult to prove’. What our eye perceives of our physical surrounding is in reality nothing more than a succession of colours and it is only the interpretation made by our brain which gives a meaning to these images. Therefore, we do not see what we think we see. However, it is assumed that an external physical world exists independent of our own existence. Boyce et al. [21] finally note that ‘what we seem to mean by direct observation is that statements can be made concerning our sensations using terms whose application is easily and directly verifiable by multiple observers’.

Indirect Measurement *Indirect measurement* is typically used to measure objects and phenomena that are difficult to observe in a direct manner. Hence, measurement is achieved through inference from another observable phenomenon. Indirect measurement assumes the existence of a connection between the object or phenomenon under consideration and another, directly observable object or phenomenon. It is then from the observation of the connected phenomenon that the measurement of the first is inferred. For example, temperature can be measured by first measuring the pressure of the air, since pressure and temperature are known to be directly related. Likewise, a planet is inferred to be orbiting a distant star on the basis of periodic changes in the observed radiated light from the star. The planet is not seen directly, it is observed indirectly.

Opinion Measurement Human behaviour can be observed and recorded by studying the processes and activities that humans carry out. However, some aspects cannot easily be measured from direct observation. *Opinion measurement* is concerned with these aspects, which include the subject’s opinions, their feelings, beliefs, and thought processes. These aspects cannot be observed directly, but the subject’s expression of it can be observed and recorded. Questionnaires, surveys and polls are typical instruments to collect such data. A questionnaire is a structured list of questions, administered in person by an investigator, by telephone, or by mail. A survey is an inquiry which involves the collection of systematic data across a sample of random cases and the statistical analysis of the results. Finally, an opinion poll is similar to a survey, but its purpose is not that of scientific inquiry, but rather the prediction of attitudes and feelings that a population will have concerning various topics, events, and personalities. The opinion poll is a technique that is typically used in politics and in marketing.

2.1.4. Measurement Theory

Measurement is an essential part of disciplines such as the social, economical or behaviour sciences. But contrary to other disciplines such as physics, a theoretical foundation for measurement has long been missing. For example, how can air pollution be measured with one index that takes account of many different pollutants? Is it meaningful to assert that the customer price index has increased by 20%? And how can complex concepts such as brightness, preference or intelligence be measured? *Measurement theory* is a large body of research work, stimulated by the need to put measurement on a firm foundation. Roberts [105] notes that measurement theory 'seems to have no disciplinary home'. Indeed, much of the work has been produced by mathematicians, physicists, economists, psychologists, and others. Measurement theory acts as a foundation for measurement in the social, economical and behaviour sciences.
According to Roberts [105], early influential books on measurement were written in the 1920's and 1930's by Campbell N. R., Cohen M. R., Nagel E., and Guild J. This literature is concerned with measurement in the physical sciences. However, this approach to measurement was not broad enough to encompass the problem of measurement in the social sciences. A broader approach to measurement than that of the classical writers started to appear in the 1950's and 1960's. Authors such as Stevens S. S., Scott D., Suppes P., Zinnes J., Pfanzagl J., and Krantz D. H. shaped the fundamental concepts of measurement theory.

In the following three sections, we discuss some of the fundamental issues in measurement theory, notably fundamental measurement, scales and derived measurement. Our aim is to provide a brief overview of the subject, the reader interested in further details is referred to Pfanzagl [99], Roberts [105], and Suppes and Zinnes [111]. Fenton [35] discusses measurement theory in the context of software measurement.

2.2. Fundamental Measurement

Pfanzagl [99] states that 'it is the properties [of objects] which are the concern of measurement, and not the objects themselves'. Weight, eye colour and intelligence are typical examples of properties of persons. In measuring one property, we neglect all the other properties the objects in question might have. For example, in measuring weight, we neglect other properties of the object such as shape and colour. Thus, different objects might become equivalent if consideration is restricted to one property.
Furthermore, a property may have a distinct structure. According to Pfanzagl [99], this structure is 'determined by empirical relations between empirical objects'. For example, if we consider the relation between two objects regarding their weight property, we may say that one object is heavier than the other. Such relations give a specific structure to a property.
As to the measurement process, Roberts [105] notes that the central idea of represen-

tation, 'the translation of "qualitative" concepts such as relations and operations into appropriate "quantitative" (or other concrete) relations and operations enjoying known properties'. Thus, it is the numerical representation of the structure of the properties of objects which is the main concern of measurement. By the assignment of numbers, we seek to map an observed or empirical system to a numerical system which preserves all the relations and operations of the initial system. In measuring weight for example, we seek to assign numbers to the weight property of the objects in a way that relations such as 'heavier than' and 'lighter than', and operations such as addition or subtraction remain preserved.

In the following subsections, we introduce a formal approach to measurement, called *fundamental measurement*. Fundamental measurement is the process at which fundamental measures are defined, by mapping an observed or empirical system to a numerical system. In physics for instance, mass, temperature and volume are such fundamental measures. The subject is presented in a brief, simplified manner. For a more detailed discussion on the topic, we refer the reader to Pfanzagl [99], Roberts [105] and Suppes and Zinnes [111].

2.2.1. Formalisation

In measurement, we assign numbers to objects in a way that the observed relations and operations are preserved. In the case of temperature for example, the assignment of numbers allows to preserve the relation 'warmer than'. In the case of mass, it is the relation 'heavier than' that is preserved.
Suppose that A is a set of objects and R a binary relation on A which holds if and only if a is warmer than b. We then want to assign a number $f(a)$ to each $a \in A$ such that for all $a, b \in A$,

$$(a, b) \in R \Leftrightarrow f(a) > f(b).$$

Analogously, the measurement of preference for example is the assignment of numbers that preserves the observed relation 'preferred to'. A set A would in this case represent a set of alternatives and R a binary relation that holds if and only if a is preferred to b. The assignment of numbers $f(a)$ to each $a \in A$ allows us to preserve the observed relation 'preferred to'.

In some cases we might demand more of a measure than the simple preservation of a relation. In the case of mass for example, we want it to be additive in the sense that the mass of the combination of two objects equals the sum of their masses. Hence, the function f must not only satisfy the above stated condition, but also preserve the additive operation.
Suppose that o is a binary operation on the set A of objects that represents the combination of two objects, then the function f is a valued function that preserves the operation o, such that for all $a, b \in A$,

$$f(a \; o \; b) = f(a) + f(b).$$

However, such operations do not exist for every concept. In the case of temperature, a comparable operation does not exist, as for preference, such an operation might not necessarily make sense.

In measurement, we seek to relate a set of empirical objects to a set of numbers. The relation between the two sets is defined by a mapping function.
Consider two arbitrary sets A and B. A function f that assigns to each element $a \in A$ an element $f(a) \in B$ is called a *map* of A into B, symbolically $f : A \rightarrow B$. The element $f(a)$ is called the *value* of the function f at a, the set A is called the *domain* of f, and the set B is called the *range* of f. A map can be considered as a collection of pairs $(a, f(a))$ with an element of A and the assigned element of B.

2.2.2. Relational Systems and Homomorphisms

A *relational system* is an ordered $(p+q+1)$-tuple $S = (A, R_1, R_2, \ldots, R_p, o_1, o_2, \ldots, o_q)$, where A is a set, $R_1, R_2, \ldots, R_p$ are relations on A, and $o_1, o_2, \ldots, o_q$ are operations on A. A relational system whose set A consists of empirical objects and whose relations $R_1, R_2, \ldots, R_p$ on A are empirically determined is called an *empirical relational system*. A *numerical relational system* is one where A is the set of real numbers. For example, $(Re, >, +, \times)$ is a numerical relational set with one relation and two operations.
Measurement can be considered as the mapping of an observed or empirical relational system S to a numerical relational system N which preserves all relations and operations in S. Figure 2.1 gives a graphical representation of such a mapping. In measurement of

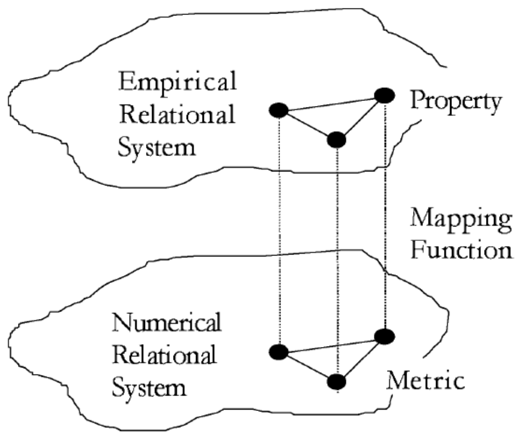


Figure 2.1.: The relational model [103]

mass, we could, in a simplified manner, consider the relational system $S = (A, R, o)$, where A is a set of objects, R is a binary relation 'heavier than' on A, and o is a binary addition operation on A. This relational system can be mapped to the numerical relational system $N = (Re, >, +)$, where Re is the set of real numbers. This relational system preserves the relation R and the operation o in S. A mapping function f from one relational system to another which preserves all relations and operations is called a *homomorphism*. Suppose $S = (A, R_1, R_2, \ldots, R_p, o_1, o_2, \ldots, o_q)$ and a second relational system of the same type $T = (B, R'_1, R'_2, \ldots, R'_p, o'_1, o'_2, \ldots, o'_q)$. A function $f : A \to B$ is called a homomorphism from S into T if, for all $a_1, a_2, \ldots, a_{r_i} \in A$,

$$R_i(a_1, a_2, \ldots, a_{r_i}) \Leftrightarrow R'_i[f(a_1), f(a_2), \ldots, f(a_{r_i})], \quad i = 1, 2, \ldots, p,$$

and, in case of a binary operation, for all $a, b \in A$,

$$f(a\ o_i\ b) = f(a)\ o'_i\ f(b), \quad i = 1, 2, \ldots, q.$$

In general, we can say that '*fundamental measurement* has been performed if we can assign a homomorphism from an observed (empirical) relational system S to some (usually specified) numerical relational system N' [105]. The measurement of temperature for example is the assignment of a homomorphism from the observed relational system $S = (A, R)$, where A is a set of objects and R is a binary relation 'warmer than' on A, to the numerical relational system $N = (Re, >)$. A homomorphism is said to give a *representation*, expressed by the triplet (S, N, f) and which is called a *scale*.
The question which arises at this point is the specification of the numerical relational system N. A relational system S can indeed be mapped to different numerical relational systems. Roberts [105] argues that the answer will depend on the theory about what is being measured and on the desired properties of the numerical assignment.

2.3. Scales

Pfanzagl [99] defines *fundamental measurement* as 'the construction of scales by mapping an empirical relational system isomorphically into a numerical relational system'. *Scales* tell us what manipulations of scale values are appropriate and lead to results that are meaningful statements about the objects being investigated.
A statement that involves a numerical scale does not necessarily remain meaningful or true if the scale is replaced by another one. Let us consider the following two statements: 'today's temperature is twice as much as yesterday's' and 'this apple weights twice as much as the other'. The first statement does not seem to make sense. Indeed, a temperature may have twice as many degrees centigrade as another, but it does not have twice as many degrees Fahrenheit. The second statement on the other hand makes sense, since the ratio of mass is the same, regardless the scale of measurement used.
In this section, we first define the notion of regular scales. We then discuss the properties of some of the more common scale types, and finally, we shall see why some statements are meaningful, while others are not.

2.3.1. Regular Scales

A *scale* is a triplet (S, T, f) where S and T are relational systems and f is a homomorphism from S to T. Given the two relational systems S and T of the same type, it is

quite possible for there to be different functions that map S homomorphically into T. Thus, a statement that involves measurement should either specify which scale (which homomorphism) is being used or be independent of scale.
Suppose f is a homomorphism from a relational system S into a relational system T, A is the set underlying S and B is the set underlying T. Suppose a function ϕ that maps the range of f, the set $f(A) = \{f(a) : a \in A\}$, into the set B. The composition $\phi\ o\ f$ is then a function from A into B. If this function is a homomorphism from S into T, we call ϕ an *admissible transformation of scale*.
Suppose two scales (S, T, f) and (S, T, g) that map a relational system S homomorphically into T, A being the set underlying S and B the set underlying T. It may be possible to find a function $\phi : f(A) \rightarrow B$ so that $g = \phi\ o\ f$. If for every scale (S, T, g), there is a transformation $\phi : f(A) \rightarrow B$ such that $g = \phi\ o\ f$, we call (S, T, f) a *regular scale*. Roberts and Franke (mentioned in [105]) characterise a regular scale as following:

> (S, T, f) is regular if and only if for every other homomorphism g from S into T, and for all a, b in A, $f(a) = f(b)$ implies $g(a) = g(b)$.

Furthermore, if we can map any two scales f and g into the other by an admissible transformation, we call $S \rightarrow T$ a *regular representation*. Roberts and Franke (mentioned in [105]) define the following theorem:

> If the representation $S \rightarrow T$ is regular and f and g are homomorphisms from S to T, then f is an absolute, ratio, interval, ordinal, or nominal scale if and only if g is, respectively, an absolute, ratio, interval, ordinal, or nominal scale.

In the following subsection, we discuss the above mentioned scale types more in detail.

2.3.2. Scale Types

In the previous section we have seen that a representation $S \rightarrow T$ is regular if we can map any two scales f and g into the other by an admissible transformation. The class of admissible transformations can now be used to define different *scale types*. Table 2.3 lists some of the more common scale types. They are classified from 'strongest' to 'weakest' in the sense that absolute and ratio scales contain much more information than ordinal or nominal scale. Furthermore, they allow more extensive analysis than the latter ones. It is therefore important in measurement to obtain the strongest scale possible in order to preserve a maximum of information.
Other scale types than the above mentioned do exist, for example the log-interval scale (admissible transformations: $\phi(x) = \alpha x^{\beta},\ \alpha, \beta > 0$) or the difference scale (admissible transformations: $\phi(x) = x + \beta$). However, these scales are less common.
In the remainder of this section, we briefly discuss the properties of the above mentioned scale types. Scales are ordered hierarchically, meaning that each higher-level

Admissible Transformations	Scale Type	Example
$\phi(x) = x$ (identity)	Absolute	Counting
$\phi(x) = \alpha x, \quad \alpha > 0$ Similarity transformation	Ratio	Mass Temperature (Kelvin) Time (intervals) Currency ($, €, ...)
$\phi(x) = \alpha x + \beta, \quad \alpha > 0$ Positive linear transformation	Interval	Temperature (Fahrenheit, centigrade) Time (calender)
$x \geq y \quad \text{iff} \quad \phi(x) \geq \phi(y)$ (Strictly) monotone increasing transformation	Ordinal	Preference Hardness Air quality
Any one-to-one ϕ	Nominal	Telephone numbers Numbers on uniforms Classification numbers of library shelves

Table 2.3.: Common scale types (based on [105])

scale possesses all properties of the lower ones. Starting with the ‘weakest’ scale type—the nominal scale—all consecutive scales will possess the properties of the preceding ones.

Nominal Scale *Nominal scale* represents the most simple operation in science, which is classification. In nominal scale, numbers are assigned to objects only as symbolic names or codes, they imply no order or relationship between the objects. Pfanzagl [99] notes that in this situation, it is not necessary to use numbers to represent such a crude structure, words are wholly sufficient for this purpose. Indeed, the only information a nominal scale provides is whether two elements $a_1, a_2 \in A$ are equivalent or not. Therefore, all one-to-one functions ϕ define admissible transformations. Any transformation changing the names in such a way that objects with different names get different names and objects with identical names get identical names leads to a nominal scale again.
For example, the numbers on the uniforms of football players are of nominal scale, or the classification numbers for library shelves. The actual numbers have no significance, they serve the sole purpose of identification of an element of a set.
Boyce et al. [21] finally argue that ‘whether the use of a nominal scale is really characterization of some attribute of an object in a quantitative manner is perhaps open to question’. Indeed, the numbers assigned have no characteristics of quantities beyond identity and difference. The usefulness of a nominal scale in measurement is therefore at best severely limited.

Ordinal Scale Scales that are unique only up to order are called *ordinal scales*. Objects can be compared within an order, but the scale offers no information about the importance of the difference between the objects. The admissible transformations are monotone increasing functions, that is, functions that satisfy the condition that $x \geq y \Leftrightarrow \phi(x) \geq \phi(y)$.

For example, in a customer satisfaction survey using a five-point scale we cannot tell how much greater 5 is than 4, nor can we say that the difference between 2 and 3 is equal to that between 5 and 6. We only know that $5 > 4$, $4 > 3$, and so on. Any numbers could be used as long as the order is preserved.
Mathematical operations such as addition, subtraction, multiplication, and division cannot be used with ordinal scales. However, as points out Kan [61], the assumption of equal distance is made in some cases and operations such as average are applied to ordinal scales.

Interval Scale An *interval scale* indicates not only the order, but also the exact distance between two objects. The numbers representing the objects indicate the magnitude of distance between the ordered objects. Interval scale is a scale with positive linear transformations as its class of admissible transformations. Such functions are of the form $\phi(x) = \alpha x + \beta, \alpha > 0$.
Temperature (Fahrenheit, centigrade) is an example of an interval scale. Given two different temperature values, we can tell that one is 'hotter than' or 'colder than' the other. Furthermore, we can tell that one value is n degrees higher or lower than the other. To change the unit of temperature, we vary the zero point (changed by β) and the unit (changed by α). To change from Fahrenheit to centigrade, we would take $\alpha = 5/9$ and $\beta = -160/9$. Time on a calender (year) is another example of an interval scale.

Ratio Scale Interval scales become *ratio scales* when an absolute or non-arbitrary zero point can be located and when the intervals are equal but arbitrary. The class of admissible transformations of ratio scales are similarity transformations of the form $\phi(x) = \alpha x, \alpha > 0$.
Mass, for example, is a ratio scale since a non-arbitrary zero point can be located. Changing the unit of mass is achieved through multiplication by a positive constant. By multiplying by 1000, we change, for example, from grams to kilograms. Intervals of time (hours, minutes, seconds, etc...) define other ratio scales.
The advantage of the ratio scale is that the full range of mathematical operations can be used on our assigned numbers with meaning. For example, we may say that a piece of wood that measures 2 meters in length is twice as long as another that measures 1 meter. Moreover, if we glue the two pieces together in length, we produce a piece of wood that measures 3 meters in length.

Absolute Scale *Absolute scale* is the most simple example of a scale. The only admissible transformation is $\phi(x) = x$, which means that there is only one way to measure things. Counting is an example of an absolute scale.

2.3.3. Meaningful and Meaningless Statements

In the introduction to this section we have seen that some statements make sense, while others do not. For instance, the statement ‘this apple weights twice as much as the other’ seems to make sense, while ‘today’s temperature is twice as much as yesterday’s’ does not. Indeed, one can always perform mathematical operations on numbers. But can one still deduce true, meaningful statements about the objects being measured after having performed such operations?

This leads us to the *meaningfulness (or meaninglessness) of statements*. According to Roberts [105], ‘a statement involving numerical scales is *meaningful* if its truth (or falsity) remains unchanged if every scale (S, T, f) involved is replaced by another (acceptable) scale (S, T, g)’. If we consider that all scales come from regular representations, we may say that ‘a statement involving numerical scales is meaningful if and only if its truth (or falsity) remains unchanged under all admissible transformations of all the scales involved’. Meaningful statements are unambiguous in their interpretation and say something significant about the fundamental relations among the objects being measured.

Let us consider the following statement

$$f(a) = 2f(b),$$

where $f(a)$ is a quantity assigned to a, for example its mass or temperature as in the above mentioned statements. A statements is meaningful if its truth (or falsity) is preserved under all admissible transformations ϕ, such that

$$f(a) = 2f(b) \Leftrightarrow (\phi \; o \; f)(a) = 2[(\phi \; o \; f)(b)].$$

If ϕ is a similarity transformation as in ratio scales, then we have

$$f(a) = 2f(b) \Leftrightarrow \alpha f(a) = 2\alpha f(b).$$

The statement $f(a) = 2f(b)$ is thus meaningful if the scale f is a ratio scale, as in the measurement of mass. The statement ‘this apple weights twice as much as the other’ remains true whether we reason in tons, kilograms, or grams.

If the scale is an interval scale, as it is the case in measurement of temperature, the admissible transformation takes the form $\phi(x) = \alpha x + \beta, \; \alpha > 0$. Generally, we find that

$$f(a) = 2f(b), \text{ but } \alpha f(a) + \beta \neq 2[\alpha f(b) + \beta].$$

Our statement is generally meaningless if f is an interval scale. The statement ‘today’s temperature is twice as much as yesterday’s’ does not make sense since its truth does not remain unchanged whether we reason in Fahrenheit or centigrade.

Roberts [105] concludes that ordinal scales can be used to make comparisons of size, like

$$f(a) > f(b),$$

interval scales to make comparisons of difference, such as

$$f(a) - f(b) > f(c) - f(d),$$

and ratio scales to make more quantitative comparisons, like

$$f(a) = 2f(b) \quad \text{and} \quad f(a)/f(b) = \lambda.$$

2.4. Derived Measurement

Derived measurement takes place when some concepts have already been measured, and new measures are defined in terms of existing ones. For example, density is a derived measure, defined in terms of mass m and volume v, as $d = m/v$. Area is expressed in terms of length l and width w, as $a = l * w$. Derived measurement are not measured fundamentally, but are derived from other scales, which may or may not be fundamental.

According to Roberts [105], no generally accepted theory of derived measurement exists. Some writers argue that derived measurement is not even measurement at all. Pfanzagl [99], for example, argues that if a property measured by a numerical scale had any empirical meaning of its own, it would also have its own fundamental scale. Indeed, a same scale could be developed either as fundamental or derived. For example, density could be defined using fundamental or derived measurement. However, Roberts [105] notes that in practice, derived measurement is a process that is frequently used.

2.4.1. An Approach to Derived Measurement

The approach to derived measurement that we present here is based on that of Suppes and Zinnes [111].

Suppose A is a set and $f_1, f_2, \ldots, f_n$ are given real-valued functions on A. These functions are called *primitive scales*. We define a new real-valued function g on A in terms of these primitive scales, called the *derived scale*. The derived scale g and the primitive scales $f_1, f_2, \ldots, f_n$ will satisfy a certain condition $C(f_1, f_2, \ldots, f_n, g)$, and any function g satisfying this condition will be acceptable. The condition C may be an equation relating g to $f_1, f_2, \ldots, f_n$, but not necessarily. In the case of density, the condition $C(m, v, d)$ holds if $d = m/v$.

If a derived scale g is defined from primitive scales by condition $C(f_1, f_2, \ldots, f_n, g)$, we say that a function $\phi : g(A) \rightarrow Re$ is *admissible in the narrow sense* if $g' = \phi \ o \ g$ satisfies

$$C(f_1, f_2, \ldots, f_n, g').$$

The primitive scales $f_1, f_2, \ldots, f_n$ are not allow to vary. In the case of density, if m and v are not allow to vary, then d is defined uniquely in terms of m and v.

We say that ϕ is *admissible in the wide sense* if there are acceptable replacement scales $f_1', f_2', \ldots, f_n'$ for $f_1, f_2, \ldots, f_n$, so that

$$C(f_1', f_2', \ldots, f_n', g').$$

In the case of density, if m and v are replaced by other allowable scales, then d will vary. Both m and v are ratio scales that can be replace by the allowable scales $m'(a) = \alpha m(a)$ and $v'(a) = \beta v(a)$. The corresponding derived scale of density is given by

$$d'(a) = \frac{m'(a)}{v'(a)} = \frac{\alpha m(a)}{\beta v(a)} = \frac{\alpha}{\beta} d(a).$$

We say that the derived scale g is *regular in the narrow sense* if, whenever $C(f_1, f_2, \ldots, f_n, g')$ holds, then there is a transformation $\phi : g(A) \rightarrow Re$ such that $g' = \phi \ o \ g$. We say that g is *regular in the wide sense* if, whenever $f'_1, f'_2, \ldots, f'_n$ are acceptable replacement scales for the scales $f_1, f_2, \ldots, f_n$, and whenever $C(f'_1, f'_2, \ldots, f'_n, g')$ holds, then there is a transformation $\phi : g(A) \rightarrow Re$ such that $g' = \phi \ o \ g$. Thus, density is a regular scale in both the narrow and wide sense. In the narrow sense, the identity is the only admissible transformation. In the wide sense, similarity transformations are the only admissible transformations.

The scale type of a derived measurement can thus be defined in a narrow and wide sense. In the case of density, it is an absolute scale in the narrow sense, and a ratio scale in the wide sense.

Let us consider a more complicated example of derived measurement, the consumer price index, and discuss the meaningfulness or meaninglessness of a statement. This index relates the current prices of a number of basic commodities, including food, clothing, fuel, etc., to the prices at some reference time. Suppose the index is based on n fixed commodities, $p_i(0)$ is the price of commodity i at the reference time, and $p_i(t)$ is the price of commodity i at time t. The consumer price index could then be given by

$$I(p_1(t), p_2(t), \ldots p_n(t)) = \frac{\sum_{i=1}^n p_i(t)}{\sum_{i=1}^n p_i(0)}.$$

In this example, I is a derived scale that is defined in terms of $p_1, p_2, \ldots, p_n$. Each price is measured on a ratio scale and the admissible transformations in the wide sense are similarity transformations that convert a price from dollars to euros, euros to pounds, etc. But the scales are independent, so an admissible transformation of I in the wide sense results from a choice of positive numbers $\alpha_1, \alpha_2, \ldots, \alpha_n$. Thus, the statement

$$I(p_1(t), p_2(t), \ldots, p_n(t)) > I(p_1(s), p_2(s), \ldots, p_n(s))$$

is meaningless in the wide sense, since it is possible to choose the α_i so that

$$\frac{\sum_{i=1}^n p_i(t)}{\sum_{i=1}^n p_i(0)} > \frac{\sum_{i=1}^n p_i(s)}{\sum_{i=1}^n p_i(0)},$$

while

$$\frac{\sum_{i=1}^n \alpha_i p_i(t)}{\sum_{i=1}^n \alpha_i p_i(0)} < \frac{\sum_{i=1}^n \alpha_i p_i(s)}{\sum_{i=1}^n \alpha_i p_i(0)}.$$

Roberts [105] concludes that, in order to allow meaningful comparisons of the index I, all prices need to be measured in the same units. Thus, an admissible transformation of I needs to result from the multiplication of each $p_i(t)$ and $p_i(0)$ by the same positive number α. It is now even meaningful to assert that an index has doubled or increased by 20% between time s and t.

For a detailed discussion on specific derived measurements, including energy use, consumer price and air pollution indexes, we refer the reader to Roberts [105].

2.4.2. Summary Operations

Derived measurement relies on *summary operations* that allow the definition of new measures in terms of existing ones. In this section, we discuss some of the frequently used, basic operations such as ratio, proportion, percentage, and rate. Moreover, we examine the average operation, which is a measure of central tendency, and two measures of deviation, the variance and the standard deviation.

Ratio A *ratio* is a unitless quantity that denotes an amount or magnitude of one quantity relative to another. It is calculated by dividing one quantity by another, where the numerator and denominator are of the same unit, from two distinct populations and mutually exclusive. For example, in finance, the Efficiency Ratio of a business is defined as:

$$\text{Efficiency Ratio} = \frac{\text{Expenses}}{\text{Revenue}}$$

In this particular example, a lower ratio is better since it means that the earnings are much higher than the expenses. A ratio can also be written as two numbers separated by a colon. A ratio of 2:3 for example means that the whole is made up of 2 parts of one entity and 3 parts of another, the whole contains thus 5 parts.

Proportion A *proportion* is different from a ratio in that the numerator is part of the denominator. It is thus a part which is compared to the whole.

$$p = \frac{a}{a+b}$$

The proportion of satisfied customers, for example, would be defined as:

$$\text{Satisfied Customers Proportion} = \frac{\text{Number of satisfied customers}}{\text{Total number of customers}}$$

The numerator and the denominator in a proportion need not be integer values. In that case, proportions are usually called fractions.

Percentage A proportion or a fraction becomes a *percentage* when it is expressed in terms of 'per hundred units'. A proportion or fraction p is therefore equal to $100p$ percent ($100p\%$).
Kan [61] notes that enough contextual information should be given with percentages, especially what they are relative to (the total that corresponds to 100%). This allows the reader to interpret the information correctly. Furthermore, if percentages are calculated from a number of cases such as the total number of defects of a product, the number of cases should be sufficiently large. Percentages calculated from a small number of cases are not stable. Therefore, a minimum sample size of 30 should be observed for percentages, otherwise, absolute numbers should be used.

Rate Ratios, proportions and percentages that we have discussed previously are static summary measurements. They provide a view of the phenomena of interest at a specific moment in time. A *rate* on the other hand is associated with the dynamics of the phenomena of interest. It can be defined as a measurement of change in one quantity (the numerator) per unit of another quantity (the denominator) on which the former depends. The denominator is usually associated to a period of time. In demography for example, the General Fertility Rate is defined as:

$$\text{General Fertility Rate} = \frac{\text{Number of live births to women aged 15-49}}{\text{Number of women aged 15-49}} * 1000$$

It measures the number of births per 1000 women aged 15 to 45 in a given calender year.
The concept of *exposure to risk* is also central to the definition of rate. It states that all elements or subjects in the denominator have to be at risk of becoming or producing the elements or subjects in the numerator.

Mean, Average A *mean* or *average* is a measure of central tendency of a set of observations. It is calculated by dividing the sum of the values of a set of observations by the number of observations. It is usually designated by μ if we are dealing with a complete population or $\overline{x}$ if this is a sample.

$$\overline{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

The value of the mean is affected by every observation. Thus, it can be thought of as the centre of gravity of a distribution. Outlying values which are atypical of the distribution will affect the mean and move it toward the tail of a distribution.
The mean that we have just discussed is the *arithmetic mean*. There are other ways to define a mean, notably the *geometric mean* and the *harmonic mean*.
The *geometric mean*, designated $\overline{x}_g$, is sometimes used to produce an average of percentages, ratios, and rates.

$$\overline{x}_g = \left(\prod_{i=1}^{n} x_i \right)^{\frac{1}{n}} = \sqrt[n]{x_1 \cdot x_2 \cdot \ldots \cdot x_n}$$

The *harmonic mean*, designated $\overline{x}_h$, is not that commonly used. It provides a compromise value when sample size are unequal but the statistical procedures under consideration require equal sample sizes.

$$\overline{x}_h = \frac{n}{\frac{1}{x_1} + \frac{1}{x_2} + \ldots + \frac{1}{x_n}}$$

Variance *Variance* allows to characterise the dispersion of the observations, the degree to which they differ among themselves. The idea is to discover how far each

observation is from the mean and to find the centrality of these differences. However, since the sum of the values of the deviations will always be zero, so will their average. To solve this problem, the deviations of each of the observations from the mean are put in square. Then, they are added together and divided by the number of observations.
Variance is usually represented by σ^2 for a complete population or s^2 for the sample variance. Their formulas are:

$$\sigma^2 = \sum_{i=1}^{n} \frac{(x_i - \mu)^2}{n} \qquad s^2 = \sum_{i=1}^{n} \frac{(x_i - \overline{x})^2}{n-1}$$

Standard Deviation The *standard deviation* is a related measure of dispersion. It is represented by σ for a complete population or s for a sample, and is also called the root mean square. It is defined as following:

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (x_i - \mu)^2} \qquad s = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \overline{x})^2}$$

2.5. Quality of Measurement

When measurements are being taken, one might wonder how good the measures and the resulting measurement data are. Do they really accomplish their task, which is to measure a specific object or phenomenon? And how precise and dependable are the resulting data? Numerous criteria exist that allow to evaluate the quality of measurement [20, 21, 61]. *Accuracy*, for example, expresses how well a measured value agrees with the real or standard value. *Resolution* denotes the granularity of measurement, it indicates the smallest change that can be detected. *Timeliness* captures how often data changes or how often new data is created in a source.
Kan [61] considers reliability and validity to be the two most important criteria of measurement quality.

Reliability, also called precision, refers to ‘the consistency of a number of measurements taken using the same measurement method on the same subject’ [61]. If repeated measurements are highly consistent or even identical, then the measurement method or the operational definition has a high degree of reliability. If, in the contrary, the variations among repeated measurements are large, then the reliability is low. With a measurement definition that specifies precisely such considerations as when to take the measurements, the specific scale to use and who is to take the measurements, it is likely that reliable data will be obtained. On the other hand, a vaguely defined definition may produce data with a low degree of reliability. However, the measurement of any phenomenon contains always a certain amount of error. Error-free measurement is, according to Kan [61], never attained in any discipline. The goal is therefore to minimize the measurement error in order to achieve the best possible reliability. Reliability can be expressed in terms of the size of the standard deviations of the repeated measurements. When variables are compared, usually the index of variation is used. The index of variation is a ratio of the standard deviation to the mean:

$$\text{Index of variation} = \frac{\text{Standard deviation}}{\text{Mean}}$$

The smaller the value of the index of variation, the more reliable are the measurements.

Validity refers to 'the extent to which an empirical measure reflects the real meaning of the concept under consideration' [61]. In other words, does a measure really measure what it is intended to? For simple concepts such as body height or weight, validity simply translates accuracy. However, for more abstract concepts, validity of measurement can be a difficult issue. Often, it is difficult to recognise that a measure is invalid in measuring a phenomenon, improving the measure or inventing a new one can be even more difficult.

Measurements that are reliable are not necessarily valid, and vice versa. Figure 2.2 illustrates graphically the difference between the two concepts. Given the aim of hitting

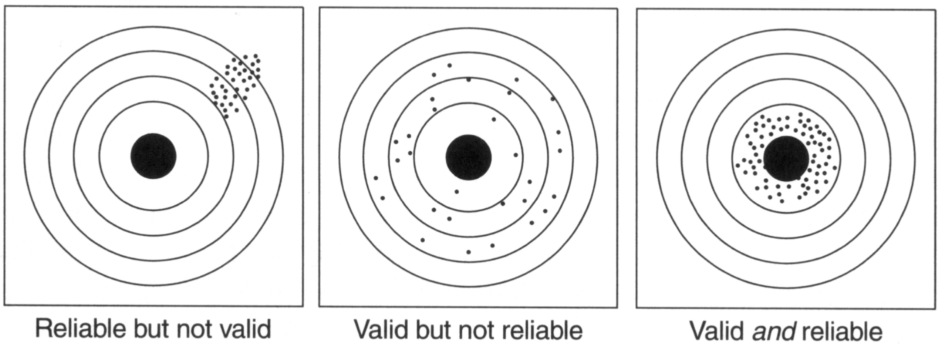


Figure 2.2.: An analogy to validity and reliability [61]

the centre of the target, which represents the object or phenomenon under investigation, reliability translates as a tight pattern, regardless of where it hits. Validity, on the other hand, is represented by hits arranged around the centre of the target.
As to the measurement errors, we can distinguish two types: *systematic* and *random errors*. Systematic measurement errors are associated with validity, whereas random errors are associated with reliability. For example, a weight scale with an offset of 5kg will, each time it is used, show a measurement that is systemically 5kg higher than the actual weight. In addition, it will show slight, random variations among measurements. Therefore, the measured value does not equal the true value because if the systematic deviation of 5kg and the random variances. In general, we can say that:

$$M = T + s + e$$

where M is the measured value, T is the true value, s is systematic error, and e is random error. Eliminating systematic error, which is associated to validity, can be achieved through better understanding of the concept of concern and though better definitions. Random error, which influences reliability, can be reduced by deriving good definitions and, based on them, by rigorous execution of measurement operations and data collection.

Kan [61] notes that there is some tension between validity and reliability. To obtain reliable data, the measurement needs to be specified precisely; the risk of being unable to represent the theoretical concept validly may however be high. On the other hand, for a definition to have good validity, is may be difficult to define the measurements precisely. It is not uncommon therefore that trade-offs between validity and reliability of measurement need to be made.

Reliability and validity are two important issues of measurement quality and they should be well thought-through before a measure is proposed, used, and analysed.

2.6. Limits of Measurement

2.6.1. Abstract Concepts

Some concepts, typically theoretical constructs with no physical existence, are difficult to apprehend since no generally accepted definition exists. Defining such concepts in respect of measurement is even more difficult.

Human intelligence is a typical example. Boyce et al. [21] note that there is no universal acceptance of the meaning of intelligence, even less of a measure for it. Scholastic aptitude tests for example do measure some form of intelligence, but they do also neglect a number of important attributes.

Quality is another example. What is quality? And how can one measure it? For some people, it represents perfection. For others, it is something that can be considered as acceptable, for still others, quality is indefinable, except that they know it when they see it. People perceive and interpret quality in different ways, and there is no assurance that two observers would agree on the relative quality of an item. Thus, the definition of quality seems to depend largely on emotional and subjective aspects. However, in many domains it is important to define quality in a way that its definition can lead to actual measurements of quality. Crosby (mentioned in [61]) defines quality as 'conformance to requirements'. This definition implies that requirements are clearly stated in a way that they cannot be misunderstood. Measurements can then be taken regularly to determine the conformance to those requirements.

Let us consider a final example: the concept of 'relevance' in the evaluation of answer sets in relation to queries in an information retrieval system. Boyce et al. [21] note that a document may be very similar as to the topic of a query, and yet not be of use to a particular information need of the user. Indeed, the user may already be aware of the document, or he may be unable to understand it sufficiently. In the latter case, the value of the document may change over time as exposure to other information allows the user to gain understanding. Thus, a same document may be considered as non-relevant by

a user at some point, and as relevant at another, making measurement of relevance a difficult task.
From these examples we can conclude that the measurement of concepts that involve subjective judgements is extremely difficult. However, quantitative methods can be used. Boyce et al. [21] argue that ‘their use involves a process of simplification which will slant the outcome with the assumptions placed upon the data, but will, in all probability, lead to a solution’.

2.6.2. Measurement and Human Behaviour

Processes and activities that humans carry out can be measured and recorded by observing their behaviour. However, observing human behaviour bears some problems. Boyce et al. [21] note that human behaviour might be affected by the knowledge that a subject is being observed. A persons who is aware that he is under observation may well behave differently than the way he behaves normally. He may behave in a way he believes he is expected to behave. First evidence of the impact of observation on measurement came from a set of experiments carried out by Elton Mayo between 1927 and 1933 at the Western Electric’s Hawthorne Plant near Chicago (described in [98]). In this study, attempts were made to measure and correlate the productivity of workers with their physical working conditions (such as degree of lighting or noise). The findings from the study were completely unexpected, as no correlations between productivity and conditions could be found. Regardless of changes in working conditions, the resultant productivity of the workers increased. Elton Mayo and his researchers finally concluded that it was their own presence that affected the outcome of the study. This phenomenon is known today as the Hawthorne effect.
Inadequately designed measures may also result in dysfunctional human behaviour [9, 88, 91]. The way a particular measure is calculated may encourage individuals to modify their behaviours in order to ensure a positive outcome, even if this involves the pursuit of inappropriate actions. Neely and Bourne [88] provide some examples of such behaviour. In call centres, the average time taken to answer the phone is often used as a measure. Operators may thus be tempted to make their line ring busy when they are particularly occupied, simply to avoid that particular calls are noted as unanswered within the target time. Alternatively, operators may pick up phones and put them back down again without ever speaking to the person at the other end, just to ensure that the call is answered within seven rings.
It is thus important to bear in mind when observing and measuring the behaviour of humans that their behaviour may be affected by the knowledge of being observed. Moreover, measures should be designed in a way that eliminates or reduces at least the opportunity of pursuing inappropriate actions simply to ensure positive outcome.

2.6.3. Fundamental Uncertainty

Boyce et al. [21] argue that there are limits beyond which we cannot measure accurately. These limits are not imposed by the measuring devices, nor by the extremely small size of certain entities that we attempt to measure, but by the way that nature presents itself to us.

The physician Isaac Newton advanced in the 17th century a number of physical laws. Newton believed that if the complexity of any system is understood, then eventually every known interaction in it can be accurately predicted. However, experimental findings in quantum physics at the beginning of the twentieth century have moved the physical sciences towards understandings that accept fundamental uncertainty at their core. These new understandings led to a set of post-deterministic principles, notably uncertainty, bounded instability and self-organisation. *Uncertainty* is grounded in two properties: sensitive dependence on initial conditions and the impossibility of measurement without participation. The first property states that accurate prediction of the behaviour of non-linear systems is impossible unless every property of every element is precisely known, whereas the second states that this knowledge is unattainable since the act of measurement influences the system under examination. *Bounded instability* describes the paradoxical existence in a system of both randomness and order. While the individual elements may appear to act randomly and with autonomy, some form of order is noticeable at the aggregate level, an order which never exceeds some finite boundaries. Changes in such bounded, non-linear systems are believed to happen suddenly and unpredictably and are thought to have the potential to cause massive shifts in the system. This process of destabilisation of the system, caused by external or internal changes and followed by reconfiguration to a new state of bounded instability, is known as *self-organisation*.

Therefore, observations will always be influenced by a number of unaccounted variables since even the act of measurement influences the system being examined. Boyce et al. [21] point out that 'if what we measure influences the instrument of measurement (as it must if we are to observe a result), then it follows that the instrument may influence that which is measured'. Thus, there is no such thing as exact measurement.

Palmer and Parker [98] further observe that many measurement models—and especially in the field of performance measurement—are largely based on deterministic assumptions about the world, while in the physical sciences, scientists have come to accept that the world has a fundamental uncertainty at its core. This uncertainty suggests that individual level measurements can be seriously flawed and that reasonable predictive patterns appear only at aggregated levels, which is where the system is in a state of bounded instability. Palmer and Parker [98] conclude that aggregation of measurements is far more useful than measures at an individual level. They argue that 'individual measurement is, and will always be, unknowable at a precise level, incapable of being measured without the observer becoming part of the system, sensitive to initial conditions, and non-predictive'. However, the collection of individual measures may be worthwhile if the focus is to aggregate the measures. Focusing on such aggregate measures may in turn initiate spontaneous self-organisation of the system.

Chapter 3.

Performance Measurement

In this chapter, we attempt to answer our second research question: What is performance measurement? What are the fundamental concepts involved, how are they related, and what approaches to performance measurement do exist?
We start which examining the terminology of the performance measurement domain and provide a brief overview of its history. We then study the core concepts involved in performance measurement, which include performance indicators, targets, and objectives. Each of the concepts is discussed in detail and examined from different points of view. Other issues related to performance measurement are then mentioned briefly. Finally, we examine the balanced scorecard framework, one of the most popular approaches to performance measurement.

3.1. Measuring Performance?

The subject of this chapter is the measurement of *performance*. But what exactly is it that we call performance? A play performed in the street, for instance, is called a performance, and stockbrokers and shareholders use the term performance when talking about the evolution of the value of particular shares. In the Oxford Dictionary of English [95] we find the following definitions: a performance is 'an act of presenting a play, concert, or other form of entertainment', or 'the action or process of performing a task or function'. Furthermore, it can designate 'a task or operation seen in terms of how successfully it is performed', a definition which we consider perfectly suitable for our subject area.
The measurement of performance is not a new subject. Measurement in fields such as finance and production has indeed a long history. What is known today as *performance measurement* is an approach that emerged in the mid 1990's and which advocates a multi-dimensional, strategy-based approach to performance. Neely [87] estimates that between 1994 and 1996, over 3000 articles on performance measurement were published, and that in 1996, new books on the subject appeared at a rate of one every two weeks in the USA alone. The more astonishing is it, that the key concepts in this field have only rarely been defined. Neely et al. [89] note that 'performance measurement is a

Author / Source	**Definition**
Neely et al. [89]	The process of quantifying the efficiency and effectiveness of action.
Dictionary of Business and Management [97]	The process of (a) developing indicators to assess progress towards certain predefined goals and (b) reviewing performance against these measures.
Wettstein [116]	The measurement, analysis and communication of performance, as well as the planning of actions and initiatives[a].
Lohman et al. [76]	The activity of measuring performance using performance indicators.

[a]Translated by the author.

Table 3.1.: Definitions of the term 'performance measurement'

topic which is often discussed but rarely defined'. Table 3.1 lists some of the definitions that can be found in literature. These definitions suggest that performance measurement is a process or an activity which involves the quantification of the results of actions and their comparison to predefined goals. Some authors further include the development of the instruments and the planning of resulting actions and initiatives in the definition.

Throughout literature, numerous terms are used to designate the instrument that actually expresses performance. Some authors call them 'performance measures', others 'performance metrics', and still others 'key performance indicators'. We use the term *performance indicator* since it is used by a great number of authors. Table 3.2 lists some of the definitions that can be found in literature. They suggest that a performance

Author / Source	**Definition**
Neely et al. [89]	(Performance Measure) A metric used to quantify the efficiency and/or effectiveness of an action.
Lohman et al. [76]	A variable that expresses quantitatively the effectiveness or efficiency or both, of a part of or a whole process, or system, against a given norm or target.
Lorino [77]	An information meant to help an individual or, more generally, a collective actor to guide the course of an action towards the achievement of a goal, or to help him to evaluate the result of the action[a].
Eckerson [31]	A metric that embeds performance targets so organisations can chart progress toward goals.
Bauer [7]	Key performance indicators are quantifiable metrics which reflect the performance of an organization in achieving its goals and objectives.
PSM [101]	An indicator is a measure or combination of measures that provide insight into an issue or concept. Most indicators compare actual values with baselines (plans or targets).

[a]Translated by the author.

Table 3.2.: Definitions of the term 'performance indicator'

indicator is a measure that quantifies the result of action against a goal. We propose the following definition for the term performance indicator: a performance indicator is a strategic instrument which allows to evaluate performance against targets. Lorino [77] adds that a performance indicator is not an ‘objective’ measure, since the measurement is not independent of the observer. In the contrary, the indicator is defined by its author in accordance to the type of action he conducts and the goals he pursues. Similarly, Wettstein [116] notes that performance cannot be expressed in a precise, absolute manner. Rather, a performance indicators provides a limited statement on the real performance. Therefore, a set of performance indicators are required to allow a thorough appreciation of performance.
Finally, the system that supports the performance measurement process is called, not surprisingly, a *performance measurement system*. Definitions for this term are rather scares, table 3.3 lists some of them. As we can see, the definitions vary largely from

Author / Source	**Definition**
Neely et al. [89]	The set of metrics used to quantify both the efficiency and effectiveness of actions.
Lohman et al. [76]	A system (software, databases, and procedures) to execute performance measurement in a consistent and complete way.
Simons (mentioned in [116])	Information systems that managers use to track the implementation of business strategy by comparing actual results against strategic goals and objectives.

Table 3.3.: Definitions of the term ‘performance measurement system’

one author to another, spanning from a simple set of performance indicators to an IT-supported strategy management system. We consider that, in theory, performance measurement systems do not require IT-support. However, in practice, performance measurement systems usually take the form of information systems that support the performance measurement process. We propose the following definition: a performance measurement system is an information system that allows to track the implementation of a strategy through the monitoring of performance indicators.

3.2. From Management Accounting to Performance Measurement

Since the 1990’s, the subject of performance measurement has witnessed an important academic and industrial interest. However, the subject is not new. The measurement of performance, an particularly financial performance, has always been an important aspect for organisations. Ghalayini and Noble [47] consider two main phases in the literature on performance measurement. A first phase that began in the late 1880’s and went through the 1980’s, and a second phase that started in the late 1980’s.
During the first phase, measurement of performance was conducted in a uni-dimensional

way, primarily in the field of finance and to a lesser degree in production and management [116]. Measurement of financial performance was known as *management accounting* and focused on measures such as profit, return on investment (ROI), return on sales (ROS), and profit per unit production. An important instrument in this field is the DuPont-Schema which decomposes the ROI in a tree-like structure. In production, non-financial factors such as productivity and quality dominated. Winslow Taylor was one of the most prominent authors in this field. A sort of performance measurement was also present in management. The Management by Objectives (MbO) model is probably the best known and most widely employed model. The MbO approach advocates that without objectives, optimal planning, decision-making, execution and controlling are not possible.

In the late 1970's and 1980's, many authors started to express a general dissatisfaction with the traditional, backward looking, accounting-based performance measurement approach. According to Bourne et al. [19] and Neely [87], traditional performance measures have been criticised for encouraging short termism, lacking of strategic focus, encouraging local optimisation, encouraging minimisation of variance rather than continuous improvement, and not being externally focused.

A second phase in performance measurement started in the late 1980's, as a result of changes on the world markets. Enterprises were facing increased competition, pushing them to reduce costs and enhance the value they deliver to their customers. In order to survive in this new environment, enterprises shifted their strategic priorities and adopted new management philosophies, notably Just In Time (JIT), Total Quality Management (TQM) and Business Process Re-engineering (BPR). These changes revealed that traditional performance measures had many limitations and that new approaches to performance measurement were required. In an attempt to overcome the shortcomings of traditional systems, structural performance measurement frameworks have been developed that encourage a 'balanced' or 'multi-dimensional' view. Furthermore, procedural frameworks have been proposed to support the process of performance measurement. For example, Keegan et al. propose a balance between internal and external measures and between financial and non-financial measures; Cross and Lynch describe a pyramid of measures which integrates performance through the hierarchy of the organisation; Fitzgerald et al. distinguish between the results and their determinants; and Kaplan and Norton [62] between the four perspectives of their balanced scorecard (discussed in section 3.7). Appendix B lists some of the proposed frameworks.

These contributions led to a new approaches to performance measurement. These approaches are based on strategy and objectives, and provide a 'balance' between financial and non-financial, internal and external, past and future performance measures. Moreover, they integrate all the factors critical to the success of an organisation.

3.3. Performance Indicators

Performance indicators are the main components of every performance measurement system. We define a performance indicator as a strategic instrument which allows to evaluate performance against targets. Performance indicators measure the vital few activities and processes that reflect the health of an organisation, contrary to plain measures that measure non-critical activities and processes. Furthermore, they align all levels of an organization and ensure that all individuals are 'marching' towards the same goals. Thus, performance indicators are more than plain measures, they are measures that incorporate a context. This context is given by strategy and objectives, as well as by targets. Therefore, an organisation will most likely have many measures, but only a few of them may qualify as performance indicators.

An organisation specialised in software development, for example, may consider the measures mean time to failure[1], defect density[2] and defect removal effectiveness[3] as performance indicators. In manufacturing, measures such as number of units manufactured, manufacturing quality and inventory turnover may qualify as vital indicators. Appendix A provides a list of commonly used measures from various domains which may or may not qualify as performance indicators, depending on the objectives.

3.3.1. Classifications

Performance indicators can be classified according to different criteria. We briefly discuss some of the classifications proposed in the literature.

Leading and Lagging Indicators

Fitzgerald et al. (mentioned in [89, 90, 116]) suggest that there are two basic types of performance indicators in any organisation: those that relate to results, and those that focus on the determinants of the results. Indicators that refer to results are usually called lagging indicators, while those focusing on determinants are called leading indicators.

Leading indicators measure activities and processes that have a significant effect on the future performance of the organisation. They measure the so-called *key drivers* of the organisation. Leading indicators provide an early indication about whether the strategy is being implemented successfully. They are either based on the current state, or on a future, planned state. Quality level, employee morale, and on-time deliveries are typical examples. Some authors argue that focusing on leading indicators will have a significant impact on lagging indicators. Fitzgerald et al. (mentioned in [116]) for instance write '(the investigation) highlights the fact that the results obtained are a function of past business performance with regard to specific determinants'. For example, increasing quality level, and high employee morale are often followed by higher customer satisfaction, and consequently an improvement of financial results.

Lagging indicators measure the results of past activities and processes of the organisation; its past performance. Typical examples of lagging indicators are traditional

[1] The average time between failures of a system.

[2] The number of defects relative to software size.

[3] The effectiveness of the removal of latent defects.

financial measures such as revenue, costs, and profits, and non-financial indicators such as market share.

Internal and External Indicators

Keegan et al. (mentioned in [89, 90, 116]) distinguish in their performance measurement matrix between indicators that measure internal aspects, and those that measure external aspects of the organisation, as shown in figure 3.1. Lynch and Cross (mentioned in

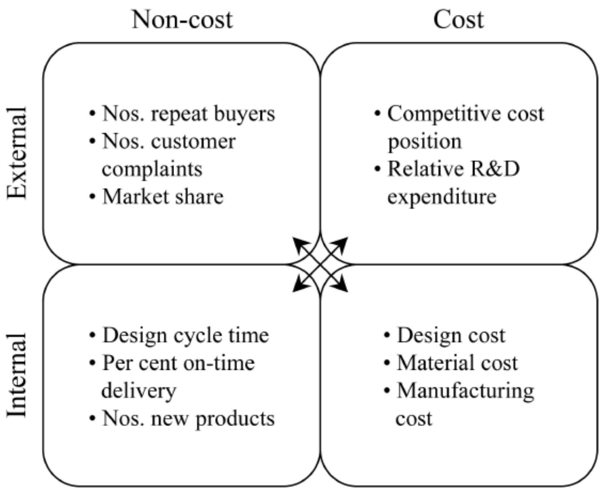


Figure 3.1.: The performance measurement matrix (Keegan et al. in [90])

[90, 116]) make also a difference in their performance pyramid between measures that are of interest to external parties and measures that are primarily of interest within the organisation, as pictured in figure 3.2. Similarly, Kaplan and Norton [63] define four distinct perspectives for measurement that represent both an external view (financial and customer perspective) and an internal view of the organisation (business process, and learning and growth perspective).
Internal indicators are typically related to processes, costs, and revenues, and may include measures such as number of new products, manufacturing costs, productivity, and cycle times. External indicators usually relate to markets, customers and shareholders and may include measures such as market share and customer satisfaction.

Financial and Non-Financial Indicators

The limitations of traditional, finance-based performance measurement have been discussed by various authors (see [19, 87]). Hence, Keegan et al. (mentioned in [89, 90, 116]) make a clear distinction in their performance measurement matrix between financial and non-financial indicators, as shown in figure 3.1. Other authors, such as Kaplan and Norton [62], also advocate the use of both financial and non-financial measures.

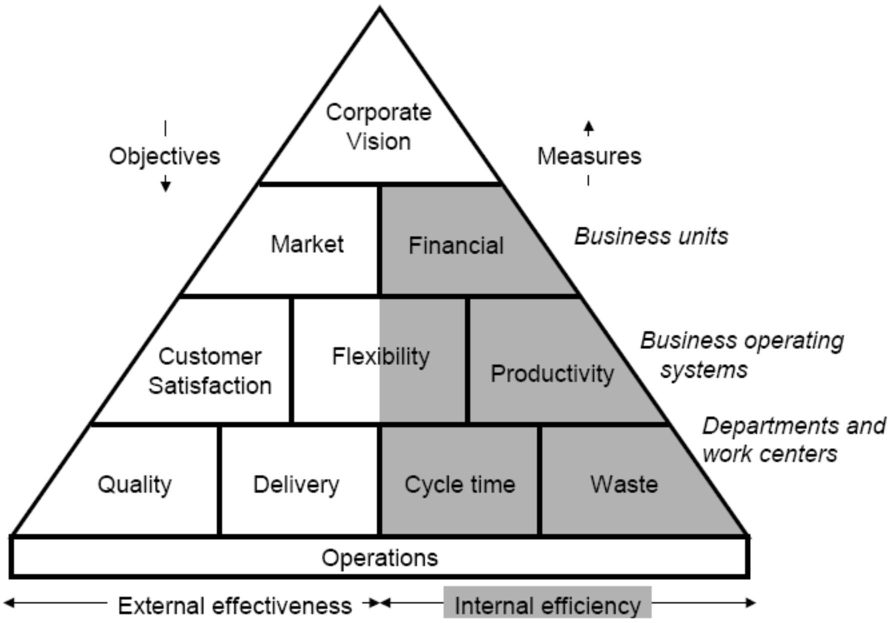


Figure 3.2.: The performance pyramid (Lynch and Cross in [116])

Financial indicators include traditional measures such as profit, return on investment and price variances. Typical non-financial indicators include quality level, customer satisfaction, on-time deliveries and employee morale.

3.3.2. Structures and Relationships

Performance indicators are rarely independent, often they are related and affect each other. A particular performance indicator may for instance synthesise several lower-level indicators, or a change in one indicator may cause a change in another. Bititci [13] notes that 'performance measurement systems should facilitate understanding of the structures and relationships between various measures and promote conscious management of inevitable conflicts'. Knowledge about the structure of and the relationships between performance indicators is valuable, since it provides a better understanding of the dynamics of the particular domain.
We present in this section a number of proposals to the structuring of performance indicators and the expression of relationships that may exist between them.

According to Bonnefous [18], performance indicators can be considered at different levels within an organisation. She defines three basic indicator types:

- strategic indicators,
- tactical indicators,

- operational indicators.

Strategic indicators are synthetic, long-term oriented performance indicators. They are designed to support executives and steering committees at the top-level of the organisation's hierarchy. Typical examples of strategic indicators include total revenues from all activities, ratio of completed projects in research and development, and satisfaction level of all clients. *Tactical indicators* are mid-term oriented and designed for managers at an intermediate level. They may include forecast accuracies of sales and inventory turnovers. Finally, *operational indicators* are short-term indicators that support the different work units in their operational activities. Number of units manufactured and percentage of products that fail quality test are typical operational indicators.
The totality of these indicators constitutes, according to Bonnefous [18], a *structured pyramid of indicators*. At the top of the pyramid, there is a limited number of strategic indicators, designed to support strategic decisions, whereas at the bottom, there is an important number of indicators that support the operational activities. The further up indicators are situated on the pyramid, the more synthetic, complex and long-term oriented they are. In the opposite, the further down indicators are situated, the more they are related to a particular activity, simple in construction and comprehension, and short-term oriented.

Lynch and Cross (mentioned in [90, 116]) propose a similar approach with the performance pyramid, shown in figure 3.2. The pyramid consists of four levels—corporate vision, business units, business operating systems, and departments and work centres—that represent the organisation's hierarchy. Objectives flow down through the organisation with a reverse flow of information and measures flowing upwards.

Suwignjo et al. [112] propose an approach that allows to identify the factors affecting performance, to structure them hierarchically and to quantify the effect of the factors on performance. The factors and their relationships are identified using cognitive maps. Figure 3.3 shows an example of the factors having an effect on 'total cost per unit'. The effects that factors have on performance are classified into direct (vertical) effect,

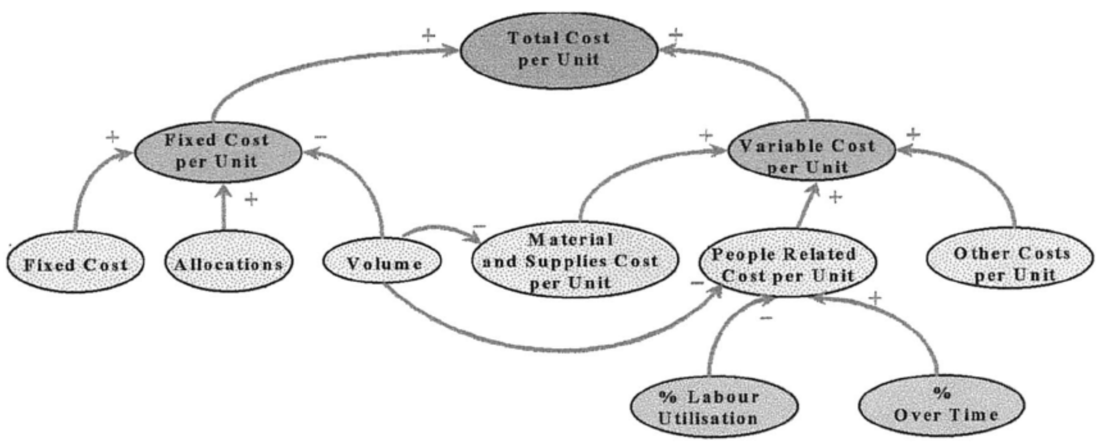


Figure 3.3.: Factors affecting 'total cost per unit' and their relationships [112]

indirect (horizontal) effect, and self-interaction effect. An effect is either positive or negative. The purpose of these maps is to uncover the factors having an effect on performance, and which, consequently, ought to be measured. The discovered factors are then structured hierarchically and the relative effects of the factors are quantified using the Analytic Hierarchy Process (AHP) procedures (described in section 3.3.3).

Kaplan and Norton [63, 64] connect the measures of the balanced scorecard through cause-and-effect relationships (described in section 3.7.2).

Popova and Treur [100] propose a formal language to specify performance indicators and their relationships, a language which is based on predicate logic. They define different types of relationships between performance indicators, among which correlation and causality. In a correlation relationship, both indicators tend to change in a similar way, whereas in a causality relationship, the change in one indicator causes the change in the other. These relationships are either positive or negative. A positive correlation indicates that both indicators increase/decrease together, whereas a negative correlation indicates that the indicators increase/decrease oppositely. The language further allows to express notions such as independency and conflict between indicators. The authors use conceptual graphs to represent these relationships graphically. Conceptual graphs have two types of nodes: concepts and relations.

Bauer [8] advocates the use of correlation analysis to understand the relationships between individual measures and to identify potential performance indicator candidates. Correlation analysis allows to measure the degree of linear relationship between two variables. The correlation coefficient may take any value between plus and minus one. The sign of the correlation coefficient—either positive or negative—defines the direction of the relationship. A positive correlation means that as the value of one variable increases, the value of the other increases, and as one decreases the other decreases. A negative correlation coefficient indicates that as one variable increases, the other decreases, and as one decreases the other increases. The absolute value of the correlation coefficient measures the strength of the relationship. A correlation coefficient of 0.8 indicates a strong positive relationship between two measures, whereas a correlation coefficient of -0.6 indicates a less powerful, negative relationship between the two measures. The author finally notes that correlation does not always indicate causation. For causation to be valid, the causal variable must always temporally precede the variable it causes.

3.3.3. Aggregation of Performance Indicators

The construction of a structured system of indicators, as described by Bonnefous [18], involves the aggregation of performance indicators along different organisational levels. *Aggregation* designates, according to Grabisch et al. [52], 'the process of combining values (numerical or non numerical) into a single one, so that the final result of aggregation takes into account in a given fashion all the individual aggregated values'. Lohman et al. [76] argue that the aggregation and the hierarchical structuring of performance indicators allows to organise them and increases insight into the cohesion between the indicators and the relationships among them. Figure 3.4 shows a hierarchical structure of performance indicators.

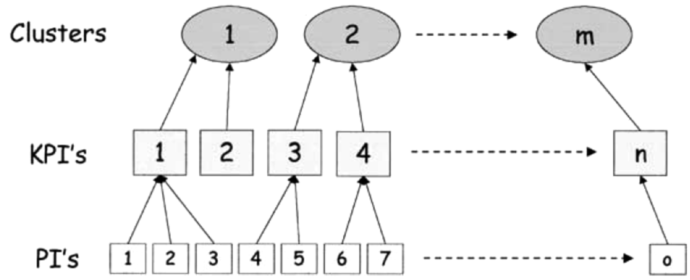


Figure 3.4.: Hierarchy of performance indicators [76]

The aggregation of performance indicators can be accomplished directly if the underlying measures are expressed in the same units of measurement, such as monetary-, product- or time-units. For example, low-level indicators such as sales by product and by location can be aggregated over different levels to a total sales for all products and all locations indicator. However, performance measurement usually involves the use of various, heterogeneous performance indicators which makes the aggregation difficult, since the units of measurement of the concerned indicators may not be identical. Folan and Browne [39] argue that one indicator could be transposed into the other. A more common approach is the calculation of the (weighted) average of the normalised values of all performance indicators [18, 76].

In this approach, the values of the performance indicators are normalised by mapping them to a linear scale, for example a 5-point or 10-point scale. Optionally, a ponderation coefficient can be defined for each indicator which translates its relative weight or importance. The ponderation coefficients are then multiplied with the corresponding, normalised indicator scores, and the average of all values is calculated. Thus, we obtain a result that takes into account all the individual aggregated values.

Rangone [104] discusses a similar approach to aggregation, called the Analytic Hierarchy Process (AHP). In this approach, the relative weight or importance of an indicator is determined through pairwise comparisons with the other indicators using a 9-point preference scale. The pairwise comparison data, organized in the form of a matrix, is then transformed into absolute priority weights. The same pairwise comparison procedure is used to normalise the actual value of each indicator to an absolute rating. To calculate the overall indicator, these absolute ratings are weighted with the absolute priority weights of each indicator before being summarised.

Other, more complex approaches to the aggregation problem do exist. Grabisch et al. [52] for instance discuss the subject of fuzzy aggregation.

Hibbard et al. [54] suggest in the context of performance measurement in the healthcare sector, that summary measures are useful. However, they argue that the degree to which such measures would be viewed as trustworthy and valid would need empiric assessment. Similarly, Lohman et al. [76] note that many conceptual questions are still not answered, questions such as: What are the effects of combining several measures into an overall score?

3.4. Targets and Ratings

Performance indicators should relate to specific *targets* [27, 49, 51, 64]. Indeed, the only way to evaluate whether the value of a performance indicator is contributing to or detracting from the achievement of an objective is by having targets associated to it. Targets represent values the organisation would like to reach at a specific moment in time. They can represent planned values, norms, expected values, or constraints [101]. Targets should be clearly stated and objectively measurable, in order to allow organisations to track progress towards objectives.
Setting appropriate target values is a critical step which should be based on historical data and forecasts. Moreover, targets should be a source of motivation for the whole organisation and, in that sense, they should neither be too easy nor too hard to achieve.

3.4.1. Static and Dynamic Targets

Globerson [49] identifies two approaches to target setting: the static and the dynamic approach. The static approach fixes the target at a certain performance level which remains unchanged. The dynamic approach expresses the target as a rate of expected improvement. Therefore, the target level in using the dynamic approach changes gradually. In a dynamic approach to target setting, a target may reflect an end-state that the organisation would like to reach, or an intermediate state—a so-called milestone target—for reaching this end-state.
While a more or less constant progression of performance is possible in some domains, others may experience important fluctuations over time due to trends, cycles and seasonality. In the hospitality industry for example, the tourist demand fluctuates dramatically from month to month due to vacation cycles, major holidays and weather profiles. Consequently, performance targets need to be adjusted to reflect this uneven customer demand over the year. To solve this kind of problem, Bauer [10] proposes a solution that 'de-seasonalises' the actual performance indicator values. Calculated from the monthly performance indicator values of the previous three years, seasonality indices allow to adjust the actual monthly values. The adjusted values reflect then the seasonal variations and can be compared to a constant performance indicator target.

3.4.2. Ratings

In order to provide a quick and easy to understand reading of whether the value of a given performance indicator is good or bad, performance indicators should be associ-

ated to a *rating* or *grading system* [11, 30]. Composed of a range of thresholds, a rating system helps qualify the gap between the actual and the target value of a performance indicator. Such a rating system can use letter grades (A, B, C, D, E), score names (Exceptional, Very Good, Good, Needs Improvement, Unacceptable) or colours (green, yellow, red) to qualify the different levels of performance. For example, indicator values that reach 90% or more of their target receive an A grade, 80%-89% of target is a B, 70%-79% of target is a C, and so on.
Sayal et al. [107] use so-called taxonomies to classify instances of a process depending on its characteristics. A taxonomy on process duration, for example, may provide a qualitative indication of the process execution time with four categories: fast (processes that last less than 5 days), acceptable (between 5 and 10 days), slow (between 10 and 15 days), and very slow (more than 15 days).
In the context of process monitoring, Bauer [9] proposes the use of control charts to facilitate the development of thresholds. A control chart is basically a group of statistical data—the performance indicator data points—plotted on a time sequence. The points are then compared to the control limits—the thresholds—in order to determine if the data is within control or not. Figure 3.5 shows the application of a control chart to the values of the performance indicator reservation wait time. The 'x double bar'

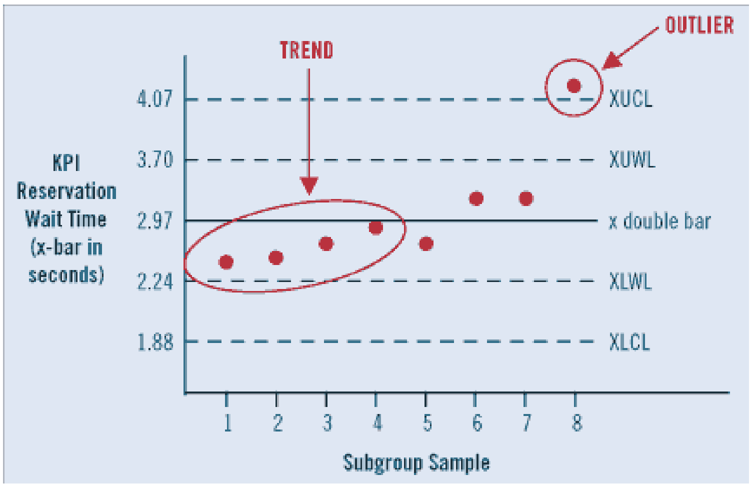


Figure 3.5.: Performance indicator control chart [9]

represents the target, whereas XUCL (Upper Control Limit), XUWL (Upper Warning Limit), XLWL (Lower Warning Limit) and XLCL (Lower Control Limit) represent the thresholds. Both the target and the limits are calculated from a set of historical indicator values using statistical operations. The control and warning limits help in identifying indicator values that are out of control, experiencing problematic trends or are within the allowable variability. Furthermore, statistical pattern detection rules can be employed to indicate the current state. The selection of appropriate threshold levels is a critical process. Bauer [9] points out that the selection of too tight thresholds can cause an overreaction to performance values that are within acceptable limits. Alternatively, too broad thresholds will not be sensitive enough to detect significant values that are outside acceptable limits.

3.5. Strategy, Objectives and Initiatives

One of the most often stated recommendation in performance measurement, notably in [49, 51, 62, 74, 79, 88], is to derive performance indicators from *strategy* and *objectives*. The Oxford Dictionary of English [95] defines a strategy as 'a plan of action designed to achieve a long-term or overall aim', whereas an objective is 'a thing aimed at or sought; a goal'. The Balanced Scorecard Collaborative [3] proposes the following definition to the term objective: an objective is 'a concise statement articulating a specific component of what the strategy must achieve or of what is critical to its success'.
Strategy and objectives must indeed be measurable somehow in order to be able to track progress. Roush and Ball (mentioned in [51]) argue that 'a strategy that cannot be evaluated in terms of whether or not it is being achieved is simply not a viable or even a useful strategy'. Thus, performance indicators should be derived from strategy and objectives to allow their evaluation. Moreover, all performance indicators should be linked to strategy and objectives. Bonnefous [18] and Holmberg [55] point out that not doing so may cause local sub-optimisations which might lead the organisation in different directions.
Manoochehri [79] finally notes that 'the purpose of performance measures is to guide and monitor employees to achieve the company's goals'. Thus, deriving measures from the organisation's strategy and objectives allows to translate them into measurable terms, able to pull people towards the overall vision of the organisation.

3.5.1. From Vision to Performance Indicators

Numerous procedural frameworks have been proposed by researchers that describe the process of translating the organisation's strategy into concrete performance indicators, appendix B lists some of them. Most of these frameworks proceed in a similar, top-down approach, as pictured in figure 3.6.
Bonnefous [18] for example proposes the following, relatively simple procedure:

1. Clearly define the strategy and deploy it in the form of objectives that need to be achieved, and this at every level of the organisation.

2. Identify the key processes and activities of the organisation that need to be improved in the context of the objectives.

3. Identify the critical success factors, those elements that have an influence on the performance of given activities and processes.

4. Choose performance indicators that measure the outcomes, but also the dynamics of progress.

5. Observe attentively the evolution of the performance indicators.

Figure 3.6.: Strategic alignment pyramid [7]

Translating the organisation's mission and vision into strategies, objectives, critical success factors and finally performance indicators ensures that all levels are strategically aligned.
An alternative approach is to proceed in a bottom-up fashion, where indicators are selected from the collection of already existing measures. However, such an approach focuses only on those measures that are already being tracked, which may exclude other measures critical to the organisation's objectives.

3.5.2. Structures and Levels

Once the overall strategy has been clarified, it can be deployed in the form of objectives. These objectives can then be related to one another and/or decomposed into sub-objectives [12, 32, 33, 63, 69, 88].
Neely and Bourne [88] propose so-called success maps to illustrate the relationships that exist between the different objectives. Similar to the approach proposed by Kaplan and Norton [63] (described in section 3.7.2), objectives are connected through cause-and-effect relationships. Figure 3.7 shows a basic example. A manufacturing company might argue, for example, that improving operating efficiency is necessary. This objective can be achieved by improving on-time deliveries, which, in turn, can be achieved by reducing lead times and improving stock control. Each of these statements acts as a lever on the achievement of the overall objective. The action-profit linkage model by Epstein and Westbrook [32] works in a similar way. Causal links between actions from four components (company actions, delivered product/service, customer actions and economic impact) are developed, which allows then to estimate the impact of specific actions on the overall profitability of the organisation.
Other authors, such as Berrah [12], Eriksson and Penker [33], and Kueng [69], advocate a hierarchical decomposition of objectives, which results in a tree-like structure. Objec-

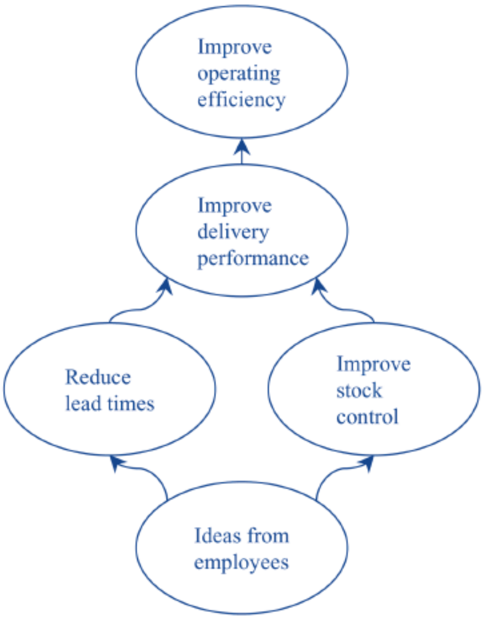


Figure 3.7.: Success map [88]

tives are decomposed iteratively into sub-objectives and elementary objectives through a cause-and-effect reasoning. Figure 3.8 shows an example. Here, the global objective ‘reduce delays’ is decomposed into three lower-level objectives—reduce delivery delays, reduce manufacturing cycle times, and reduce packaging delays—each of which contributes in achieving the global objective. These objectives are in turn decomposed into lower-level objectives, and so on. Thus, the objectives of a given level are the means to meet the objectives of a superior level.
For each objective, one or several performance indicators with associated targets can then be defined which reflect the extent to which the objective has been fulfilled. In order to measure the objective ‘increase customer satisfaction’ for example, a customer satisfaction index could be used as performance indicator. Kueng [69] notes that it may not always possible to find indicators which clearly relate to a given objective. In this case, a further refinement may solve the problem.

An objective usually applies to a specific level within an organisation. Berrah [12] defines the following three levels:

- strategic objectives,
- tactical objectives,
- operational objectives.

Strategic objectives relate to the global strategy of the organisation and are concerned with the evolution of the organisation, its orientations, and its positioning in the environment. *Tactical objectives* relate to the resources and processes of the organisation,

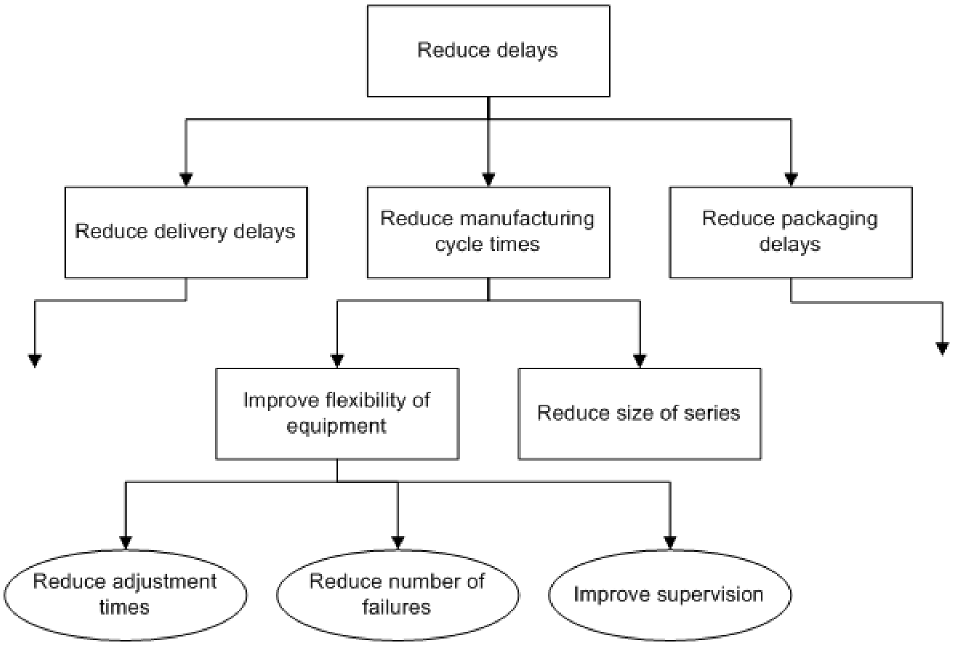


Figure 3.8.: Hierarchical decompositions of an objective (based on [12])

whereas *operational objectives* concern to the actual execution of processes.

3.5.3. Quantitative and Qualitative Objectives

We can distinguish two types of objectives: *qualitative* and *quantitative objectives*. Eriksson and Penker [33] note that 'a quantitative goal can be described with a target value in a specific unit of a measurement, while a qualitative goal is described more loosely, without a specific target value'. Determining whether an objective has been fulfilled or not can be achieved in the case of a quantitative objective through the comparison of the actual value of measurement to the target value. In the case of qualitative objectives, the evaluation relies mainly on human judgement.
High-level objectives are usually of qualitative nature, the objective 'become market leader in 5 years' is a typical example. On the other hand, the deeper in the objective hierarchy, the more specific and quantified the objectives become. 'Reduce delivery delays' or 'reduce manufacturing cycle times' are typical examples of lower-level objectives that can be quantified.

3.5.4. Initiatives

Performance measurement should stimulate *actions* [49, 64, 88, 89]. Neely and Bourne [88] state that 'the whole process of measuring performance is completely wasted unless action is taken on the performance data that are produced'. Indeed, why bother

about measurements if their results do not have any influence on the way the organisation conducts its business. Therefore, actions should be initiated and/or changes should be introduced in case of undesired deviations of particular performance indicators.
The Balanced Scorecard Collaborative [3] define *initiatives* as 'key action programs developed to achieve objectives or close gap between measures performance and targets'. Initiatives are also referred to as projects, actions, or activities. Once targets have been established for performance indicators, one can assess whether current initiatives help achieving those targets. If this is not the case, new initiatives may be required. According to Wettstein [116], initiatives have similar characteristics as projects. Notably, they are limited in time, usually unique, have a person responsible and dispose of certain resources.
As to the relation between performance indicators and initiatives, Kaplan and Norton [63] write that 'in general, a one-to-one correspondence between initiative and measure will not exist'. Rather, a set of initiatives may be required to achieve a set of outcomes. Similarly, Wettstein [116] notes that an initiative may have a major impact on certain objectives, but support also other objectives to a lesser degree.

3.6. Further Issues in Performance Measurement

In the preceding sections we have studied the major concepts related to performance measurement based on recommendations, studies and discussions found in literature. In this section, we state further issues and recommendations in the field of performance measurement. They are classified into four categories: structural, procedural, organisational and technical issues.

Structural Issues

Standardisation Performance indicators should rely on a standardised specification that includes all relevant attributes [30, 76].

Formula and data sources Performance indicators should be based on an explicitly defined formula and source of data [49].

Ownership Performance indicators should be 'owned' by individuals or groups who are accountable for their outcome and responsible for developing their methodologies [76, 79, 101, 109].

Grouping Performance indicators should be grouped according to specific subject areas or points of view [62, 76, 101].

Procedural Issues

Improvement Performance measurement should focus on trends and improvements [27, 49, 72]. Rates, ratios and percentages should be employed rather than absolute numbers.

Balance A set of performance indicators should provide a ‘balance’ between financial and non-financial, internal and external, and leading and lagging measures [30, 62].

Relevance Performance indicators should be relevant to the user’s needs [79].

Simplicity Performance indicators should be straightforward and easy to understand [30].

Discussion Performance indicators should be selected through discussions with the people involved [49].

Limited Number Performance indicators should be limited in number since having too many measures may confuse the users who may not know the relative importance of the individual measures [63, 79].

Aggregation Performance indicators should focus on aggregate-level measures since individual-level measures may be flawed due to fluctuations and randomness [49, 98].

Revision Performance indicators and targets should be reviewed periodically, modified as needed and discarded if necessary [19, 41, 48, 64, 79].

Organisational Issues

Control Performance indicators should be under the control of the evaluated organisational unit [49].

Data Collection Performance indicators should use data which are automatically collected as part of work processes [49, 69, 101, 109].

Communication The results of measurement should be communicated to relevant stakeholders [27, 88, 101, 109].

Understanding Performance indicators should be clearly understood [79, 101, 109]. Users need to understand what is being measured, why, and how their decisions and actions impact the measures.

Cost efficiency The measurement process must be cost effective to succeed [101].

Commitment Senior management commitment is essential for the successful implementation of a performance measurement system [19].

Technical Issues

Information Systems Performance measurement should rely on information systems to collect, process and report performance measures [15, 19, 79].

Timeliness and Accuracy Performance measurement systems should provide timely and accurate information [41, 49, 101, 109]. Timeliness is a relative concept which depends upon the decisions being supported.

Simple Presentation Performance information should be presented in a clear and simple form, requiring little time to be interpreted [27]. Graphical representations should be employed rather than data tables.

Information Access A performance measurement system should allow external stakeholders access to specific performance information [116].

3.7. The Balanced Scorecard Framework

In the last two decades, numerous *frameworks* have been proposed in the field of performance measurement, appendix B lists some of them. According to Folan and Browne [40], the term framework refers to 'the active employment of particular sets of recommendations', where a recommendation is 'a piece of advice'. On the purpose of performance measurement frameworks, Rouse and Putterill [106] write that they assist in the process of system building, by clarifying boundaries, specifying dimensions or views and that they may also provide initial intuitions into relationships among the dimensions.
The *balanced scorecard* is probably the most popular approach to performance measurement. A survey conducted by Marr [80] in 2004 found that out of a sample of 780 firms, 35 percent state that they follow this approach to manage corporate performance. Kaplan and Norton [63], the authors of the balanced scorecard, define it as 'a multi-dimensional framework for describing, implementing and managing strategy at all levels of an enterprise by linking objectives, initiatives and measures to an organisation's strategy'. In the following sections, we briefly discuss the main elements of this framework.

3.7.1. Four Perspectives

Introduced in 1992 by Kaplan and Norton [62], the balanced scorecard seeks to produce a 'balanced' set of measures. Performance measurement was conducted at that time with a uni-dimensional, or at least narrow focus, with an emphasis on financial accounting measures. This approach has been criticised by many authors for encouraging short termism, lacking of strategic focus, and not being externally focused. Kaplan and Norton, among others, argue that organisations should adopt a balanced set of measures, a set where financial measures balance against non-financial measures. Thus, the purpose of the balanced scorecard is to provide a clear and balanced structure of the organization's key performance dimensions.
The balanced scorecard allows to look at an organisation from four perspectives—the customer, internal business process, learning and growth, and financial perspective—and provides answers to the following questions:

- How do customers see us? (customer perspective)
- What must we excel at? (internal business process perspective)
- Can we continue to improve and create value? (learning and growth perspective)
- How do we look to shareholders? (financial perspective)

Contrary to traditional performance measurement which focuses on control, the balanced scorecard puts strategy and vision at the centre. Building a balanced scorecard involves the translation of the organisation's mission and strategy into tangible objectives and measures in each of the four perspectives. These perspectives provide a balance between financial and non-financial measures, as well as between external (customer and financial perspective) and internal measures (internal business process, and learning and growth perspective). Figure 3.9 shows the four perspectives of the balanced scorecard. We briefly discuss each of the four perspectives.

Customer Perspective Customer satisfaction has become a major preoccupation for most organisations. Thus, the mission statement on customer service is translated into specific measures that reflect the factors that matter to customers. These factors tend to fall into four categories: time, quality, performance and service, and cost. Lead time, for example, measures the time required for an organisation to meet customers' needs, from the time it receives an order to the time it delivers the product or services. Quality could measure the level of defects of delivered products, or the number of on-time deliveries. Performance and service measure how the organisation's products and services contribute to creating value for its customers. Thus, an organisation should establish general goals for customer satisfaction and then translate these goals into specific measures.

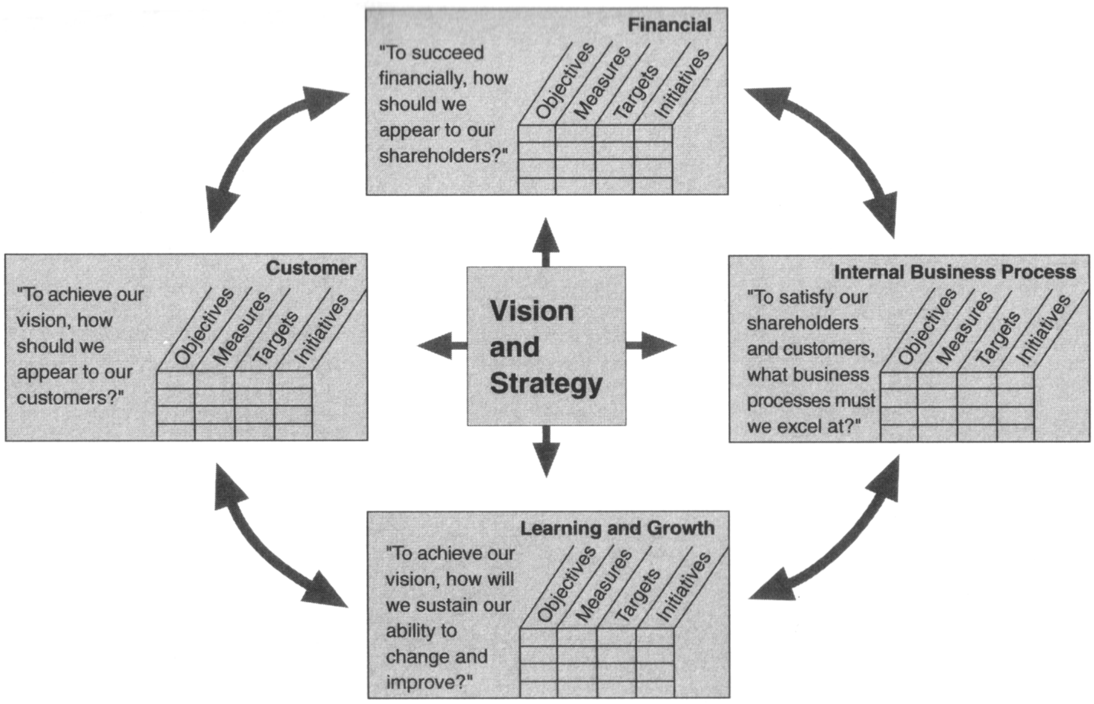


Figure 3.9.: The four perspectives of the balanced scorecard [63]

Internal Business Process Perspective Customer satisfaction derives from processes, decisions and actions occurring throughout an organisation. Measures from the customer perspective must be translated into measures of what the company must do internally to satisfy customer needs. The internal measures for the balanced scorecard should stem from the processes that have the greatest impact on customer satisfaction. These measures often relate to factors such as time, quality, productivity, employee skills and cost.

Learning and Growth Perspective Today's global competition requires organisations to make continual improvements to their products and processes, as well as the ability to launch new, innovative products and services. It is the organisation's ability to innovate, improve and learn through which it can grow and thereby increase its value. Measures from the innovation and learning perspective focus on the organisation's ability to develop and introduce new products and services, as well as improve its internal and customer process performance.

Financial Perspective The financial performance measures indicate whether the organisation's strategy, its implementation and execution are contributing to the organisations overall improvement. Typical measures in this perspective deal with profitability, growth and value.

Kaplan and Norton [63] note that a good balanced scorecard should have an appropriate mix of lagging and leading indicators. They further note that number of measures is irrelevant, as long as they support the chosen strategy. It is important however to distinguish *strategic measures* from *diagnostic measures*. Diagnostic measures simply monitor whether business is on track and can signal events that may require attention. Strategic measures on the other hand support directly the strategy of the organisation.

3.7.2. Cause-and-Effect Relationships

Kaplan and Norton [63, 64] argue that the relationships among the measures of the balanced scorecard should be made explicit so they can be managed and validated. These cause-and-effect relationships should span over all four perspectives of the balanced scorecard. Figure 3.10 gives an example.

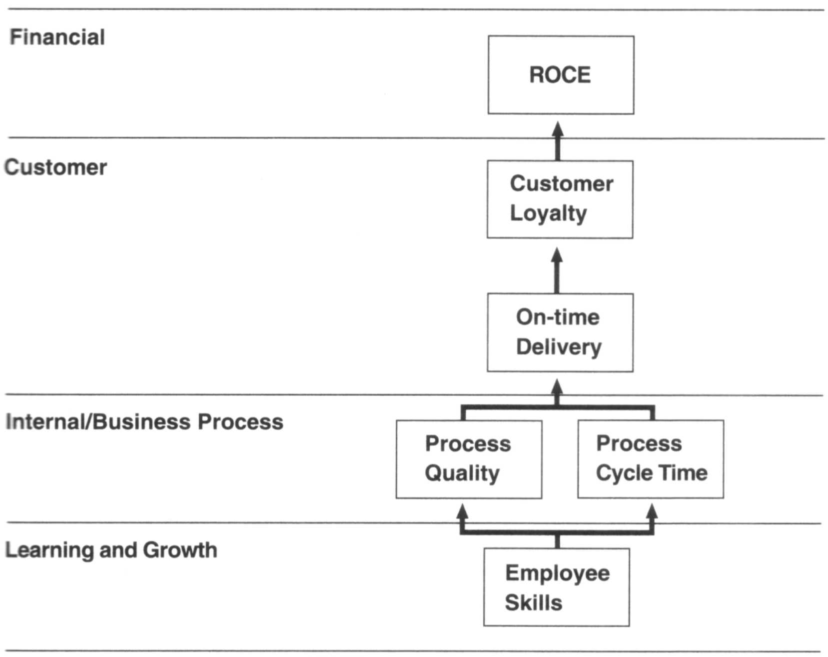


Figure 3.10.: Cause-and-effect relationships [63]

In this example, the Return-On-Capital-Employed (ROCE) is one of the scorecard measures. The driver of this measure could be repeated and expanded sales from existing customers, which, in turn, is the result of a high degree of loyalty. On-time delivery could be one of the drivers of high customer loyalty. Now, what internal processes have a significant impact on on-time delivery? Short cycle time in operating processes and high-quality internal processes may be two factors, which, in turn, can be achieved through training and improving the skills of the operating employees.
Thus, an entire chain of cause-and-effect relationships can be established across the four perspectives of the balanced scorecard that describes the organisation's strategy.

3.7.3. Four Processes

Kaplan and Norton [63, 64] added in 1996 a procedural framework to the existing structural framework. This framework consists of four processes that operate in a loop, as shown in figure 3.11. We briefly describe each of the processes.

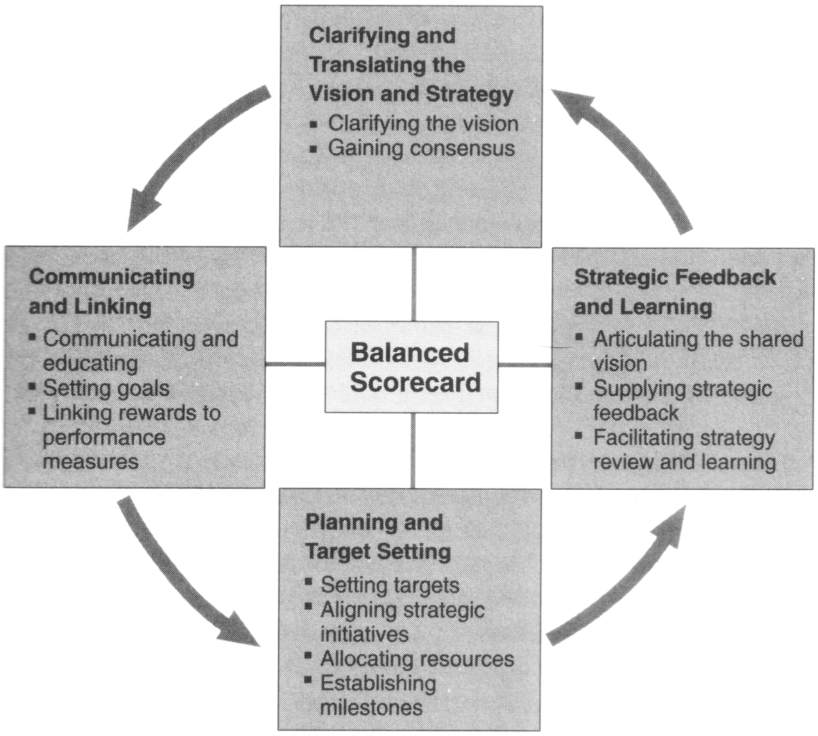


Figure 3.11.: The four processes of the balanced scorecard [63]

Clarifying and Translating the Vision and Strategy is concerned with gaining consensus on the organisation's vision and strategy and translating these statements in an integrated set of objectives and measures that describe the long-term drivers of success.

Communicating and Linking is the process by which strategy is communicated up and down the organisation and linked to departmental and individual objectives. This ensures that all levels of the organisation understand the strategy and that both departmental and individual objectives are aligned with it. Furthermore, reward systems may be linked to specific measures.

Planning and Target Setting enables the organisation to integrate their business and financial plans by allocating resources, setting priorities and coordinating initiatives that move the organisation toward its strategic objectives. Targets are established for the measures of the four perspectives.

Strategic Feedback and Learning gives organisations the capacity for 'strategic learning'. The monitoring of short-term results from the four perspectives allows the organisation to evaluate strategy in the light of recent performance, and thus enables the organisation to adjust its strategy if required.

3.7.4. Extensions and Adaptations

Kaplan and Norton have continued to extend the balanced scorecard and its performance management environment, notably with strategy development methodologies [65] and case studies.

The original balanced scorecard framework has also been adapted and extended by other researchers in order to make it respond to specific needs. Lohman et al. [76] for example propose a six-perspective model for supply chain management. Folan and Browne [39] present an extended enterprise performance measurement system which is based on four perspectives. The balanced scorecard has also been tailored to the specific needs of the Information and Communications Technology (ICT) domain. The Balanced IT Scorecard proposed by the European Software Institute (ESI) [56] for instance is designed for software producing business units. It is composed of a generic model with four perspectives and a methodology for adapting the model to the specific needs of a software producing unit. The balanced scorecard proposed by the Advanced Information Services Inc. [36], an independent software contracting organisation, expands the original four perspectives with a fifth one, the 'employee' perspective. This perspective tries to answer the question: what is important to our employees? Brock et al. [22] propose a balanced approach to IT project management, which consists of the dimensions of project, strategic alignment and program management, project processes, and project foundation.

The Balanced Scorecard Collaborative, an organisation founded by Kaplan and Norton, presented in 2000 the Balanced Scorecard Functional Standards [2], which defines the functional baseline for developing a balanced scorecard application. In 2001, the Balanced Scorecard XML Draft Standard [3] was released (described in section 4.5). Its purpose is to facilitate the exchange of performance and strategy related information between applications and across organisations.

Chapter 4.

Existing Measure and Performance Indicator Models

In this chapter, we address our third research question: What models do currently exist for measures and performance indicators?
Only few measure and performance indicator models can be found in literature. We examine seven of them. All of these models could qualify as 'generic', although some are targeted for specific domains. They are defined at different levels of granularity, and are expressed using different means of representation, varying from textual descriptions to modelling languages such as UML.

4.1. Performance Measure Record Sheet by Neely et al.

Neely et al. [91] address the simple, yet fundamental question: what does a well-designed performance measure constitute? Based on the main themes raised in the literature and on a review of recommendations, they propose a framework for specifying performance measures which 'seeks to encapsulate the elements which together constitute a "good" performance measure'. This framework—*the performance measure record sheet*—provides a structure to support the design process of performance measures. It ensures that organisations consider all of the subtle implications of the measures being proposed.
An initial record sheet, consisting of eleven elements, was tested by the authors during action research studies in various industries. The repeated applications helped identifying some shortcomings which led to some modifications to the initial record sheet. Table 4.1 shows the final performance measurement record sheet. We briefly detail its elements.

Title The title of the measure should be clear and explain what the measure is and why it is important. It should be self-explanatory and not include functionally specific

Title
Purpose
Relates to
Target
Formula
Frequency of measurement
Frequency of review
Who measures?
Source of data
Who owns the measure?
What do they do?
Who acts on the data?
What do they do?
Notes and comments

Table 4.1.: Performance measure record sheet [91]

jargon.

Purpose The rationale underlying the measure has to be specified, otherwise one can question whether it should be introduced. Typical purposes include to enable the monitoring of the rate of improvement, ensure that all delayed orders are eliminated, and ensure that the new product introduction lead time is continually reduced.

Relates to The business objectives to which the measure relates should be identified, otherwise one can again question whether the measure should be introduced.

Target An explicit target, which specifies the level of performance to be achieved and a time scale for achieving it, allows to assess whether the organisation is likely to be able to compete with others. An appropriate target for each measure should therefore be recorded. Typical targets include 20 percent improvement year on year, 15 percent reduction during the next 12 months, and achieve 98 percent on-time delivery by the end of next year.

Formula The formula—the way performance is measured—is one of the most challenging elements to specify because it affects how people behave. As a matter of fact, an inappropriately defined formula can encourage undesirable behaviours. The formula must therefore be defined in such a way that it induces good business practice.

Frequency of measurement The frequency with which performance should be recorded and reported depends on the importance of the measure and the volume of data available.

Frequency of review The frequency with which performance should be reviewed.

Who measures? The person who is to collect and report the data should be identified.

Source of data The source of the raw data should be specified. A consistent source of data is vital if performance is to be compared over time.

Who owns the measure? The person who is accountable for ensuring that performance improves.

What do they do? Actions taken by the accountable person if performance proves to be either acceptable or unacceptable.

Who acts on the data? The person who actually takes action to ensure that performance improves.

What do they do? It is not always possible to detail the action that will be taken if performance proves to be either acceptable or unacceptable, but at least the management process that will be followed needs to be defined. Typical actions include the set up of an improvement group to identify reasons for poor performance and to make recommendations.

Notes and comments Further notes and comments on the measure.

4.2. Metric Definition Template by Lohman et al.

Lohman et al. [76] present a case study on performance measurement and performance measurement systems in the context of supply chain management. The case study was carried out at the european operations department of a large company producing and selling sportswear worldwide and resulted in a dashboard prototype tailored to the needs of the company.
The theoretical contribution of their paper is to show the limitations of a 'green field approach' in the design of performance indicators—an approach that does not pay explicit consideration to already existing measures. The authors underline the importance of a 'coordination approach' focused at aligning the system with existing performance indicators. The findings of their case study point to the central role of a shared set of standardised performance indicators as a tool for achieving such coordination. This set of indicators, called the *metrics dictionary*, focuses on getting a detailed understanding of the individual measures in order to avoid failures in communication between reporting employees and managers. The *metric definition template* of the dictionary, which is based on the performance measure record sheet by Neely et al. [91], is shown in table 4.2.

Metric attribute	Explanation
Name	Use exact names to avoid ambiguity
Objective/purpose	The relation of the metric with the organizational objectives must be clear
Scope	States the areas of business or parts of the organization that are included
Target	Benchmarks must be determined in order to monitor progress
Equation	The exact calculation of the metric must be known
Units of measure	What is/are the unit(s) used
Frequency	The frequency of recording and reporting of the metric
Data source	The exact data sources involved in calculating a metric value
Owner	The responsible person for collecting data and reporting the metric
Drivers	Factors that influence the performance, i.e. organizational units, events, etc.
Comments	Outstanding issues regarding the metric

Table 4.2.: Metric definition template [76] (based on [91])

4.3. Security-metric Description of the ISM3

The Information Security Management Maturity Model (ISM3, or ISM-cubed), proposed by Aceituno [25], offers a practical approach for specifying, implementing and evaluating process-oriented Information Security Management (ISM) systems. The purpose of ISM systems is to prevent and mitigate the attacks, errors and accidents that can endanger the security of information systems and the organizational processes supported by them.

The ISM3 is based on four components: the ISM Process Model identifies the key ISM processes, the Responsibility Model provides a responsibilities-based view of the organisation, the Security in Context Model allows the tailoring of security objectives to business needs, and the Information System Model provides terminology for describing the main components and properties of information systems. The Security in Context approach aims to guarantee that business objectives are met, which is why ISM3's definition of security is context-dependent. Based on its business objectives, the organisation states its security objectives. These objectives are then used as the basis for the design, implementation and monitoring of the ISM system.

In ISM3, measures are used to determine whether security objectives are met, detect significant anomalies and to inform decisions to fix or improve the ISM processes. ISM3 measures, called *security-metrics*, are defined as shown in table 4.3.

In order to ensure that the ISM system is tailored to the needs of each environment in an organisation, the threshold set for each security target depends on the environment. A typical security and business objective could for instance state that the use of services and access to repositories is restricted to authorized users. Its corresponding security target could be set to 'fewer than two incidents every year'. When the target for a measure is set, it is compared with measured values and trends. The poor performance of a

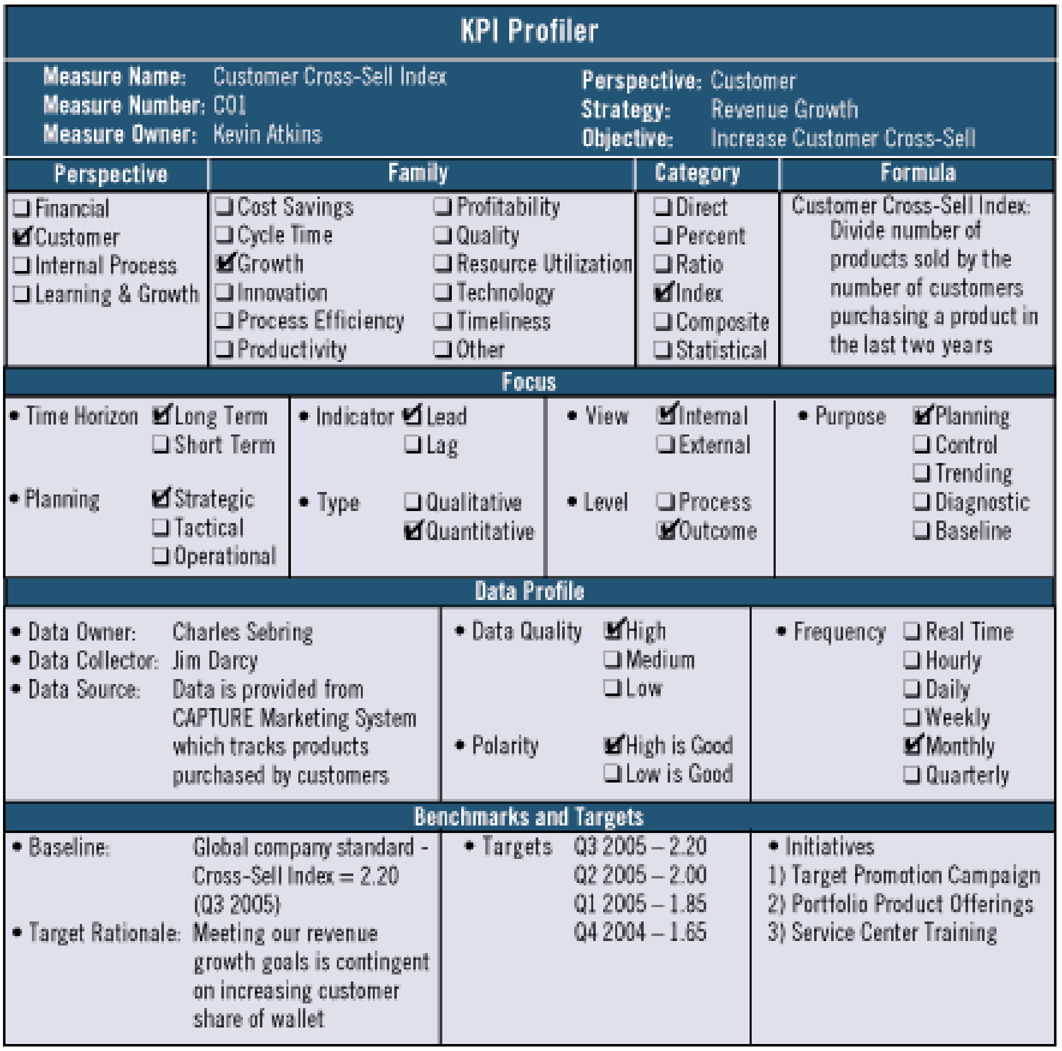
Metric attribute	**Explanation**
Metric	Name of the metric
Metric description	Description of what is measured
Measurement procedure	How is the metric measured?
Measurement frequency	How often is the measurement taken?
Thresholds estimation	How are the thresholds calculated?
Current thresholds	Current range of values considered normal for the metric
Target value	Best possible value of the metric
Units	Units of measurement

Table 4.3.: Security-metric description of the ISM3 [25]

process will take the measure outside normal thresholds. Thus, measures may be used to detect and diagnose the malfunction and take decisions depending on the diagnosis.

4.4. KPI Profiler by Bauer

Bauer [6] proposes a *Key Performance Indicator (KPI) profiler* that can be used to build, quantify and communicate KPIs throughout an organisation. The KPI profiler, shown in figure 4.1, is composed of four modules: a description module, a dimension module, a data profile module, and a benchmark and target module.
The first section of the KPI profiler, the description module, presents a high-level summary of the KPI. It includes basic information such as the name, the number and the owner of the specific KPI. Also included is information that captures the perspective the KPI belongs to and the strategy and objective it supports.
The dimension module specifies the balanced scorecard perspective a specific KPI belongs to (customer, financial, internal business processes or learning/growth perspective) and the measurement family section further specifies the category of the KPI with possible options that include cost savings, growth, process efficiency and quality. The formula defines the way the KPI is calculated and the category section specifies the unit of measurement of the output. The focus section finally gives a number of information on the nature of the KPI with indications on the time horizon, the indicator type (leading or lagging), the point of view of the measurement (internal or external) and the domain the KPI is intended for.
The data profile module indicates the data source of the KPI, the data owner and the individual in charge of data collection. Furthermore, it states the reliability of the data, whether or not high measurement values reflect good or poor results, and the frequency at which the KPI should be reported.
Finally, the benchmark and target module provides the necessary context through targets and relevant company and industry benchmarks. The baseline and the intermediate targets state values that must be reached throughout time and initiatives are activities the organisation will focus on to ensure attainment of the results.



KPI Profiler

Measure Name: Customer Cross-Sell Index
Measure Number: C01
Measure Owner: Kevin Atkins
Perspective: Customer
Strategy: Revenue Growth
Objective: Increase Customer Cross-Sell

Perspective	Family		Category	Formula
❏ Financial ☑ Customer ❏ Internal Process ❏ Learning & Growth	❏ Cost Savings ❏ Cycle Time ☑ Growth ❏ Innovation ❏ Process Efficiency ❏ Productivity	❏ Profitability ❏ Quality ❏ Resource Utilization ❏ Technology ❏ Timeliness ❏ Other	❏ Direct ❏ Percent ❏ Ratio ☑ Index ❏ Composite ❏ Statistical	Customer Cross-Sell Index: Divide number of products sold by the number of customers purchasing a product in the last two years

Focus

• Time Horizon ☑ Long Term ❏ Short Term • Planning ☑ Strategic ❏ Tactical ❏ Operational	• Indicator ☑ Lead ❏ Lag • Type ❏ Qualitative ☑ Quantitative	• View ☑ Internal ❏ External • Level ❏ Process ☑ Outcome	• Purpose ☑ Planning ❏ Control ❏ Trending ❏ Diagnostic ❏ Baseline

Data Profile

• Data Owner: Charles Sebring • Data Collector: Jim Darcy • Data Source: Data is provided from CAPTURE Marketing System which tracks products purchased by customers	• Data Quality ☑ High ❏ Medium ❏ Low • Polarity ☑ High is Good ❏ Low is Good	• Frequency ❏ Real Time ❏ Hourly ❏ Daily ❏ Weekly ☑ Monthly ❏ Quarterly

Benchmarks and Targets

• Baseline: Global company standard - Cross-Sell Index = 2.20 (Q3 2005) • Target Rationale: Meeting our revenue growth goals is contingent on increasing customer share of wallet	• Targets Q3 2005 – 2.20 Q2 2005 – 2.00 Q1 2005 – 1.85 Q4 2004 – 1.65	• Initiatives 1) Target Promotion Campaign 2) Portfolio Product Offerings 3) Service Center Training

Figure 4.1.: KPI profiler template [6]

4.5. Balanced Scorecard XML Draft Standard

The Balanced Scorecard Collaborative presented in 2001 a *Balanced Scorecard XML Draft Standard* [3] which had been elaborated together with a number of business intelligence and performance measurement systems editors. The purpose of this standard is to facilitate the exchange of performance and strategy related information between applications and across enterprises. Figure 4.2 gives a visual representation of the standard. We briefly discuss some of the key data elements.

Initiative An action program developed to achieve objectives or to close the gap between performance measures and targets. Initiatives are often known as projects, actions, or activities. Initiative information includes initiative name, description, unique identifier, an optional owner reference, start date, and end date.

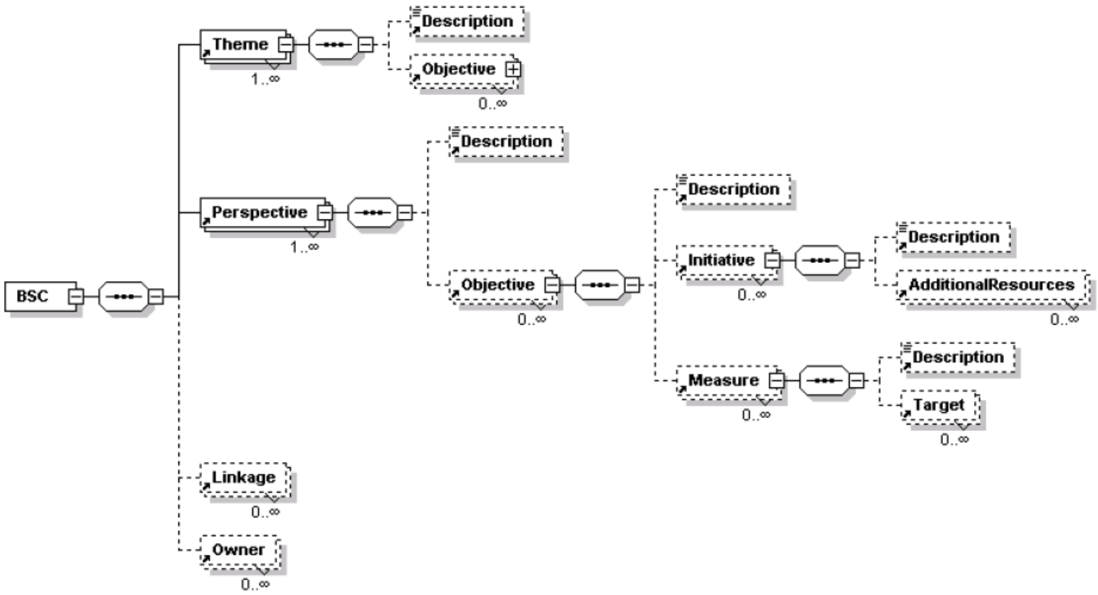


Figure 4.2.: Balanced Scorecard XML schema [3]

Measure A statement of how success in achieving an objective will be measured and tracked. Measures are written statements of what is tracked over time. Measure information includes measure name, description, unique identifier, update frequency, an optional owner reference, and associated targets.

Objective A concise statement articulating a specific component of what the strategy must achieve or of what is critical to its success. Objective information includes objective name, description, unique identifier, an optional owner reference, associated initiatives, and associated measures.

Owner Each client has a user account with which they login to access their scorecard. Owner information includes user name, unique identifier, and an optional email address.

Perspective A 'viewpoint' to a strategy as represented by key stakeholders of that strategy. Viewed horizontally, each perspective represents the set of objectives desired by a particular stakeholder (financial, customer, internal process, learning and growth). Perspective information includes the perspective name, description, unique identifier, perspective type, an optional owner reference, an optional sort order, and associated objectives.

Scorecard All the information for a client's balanced scorecard is linked back to their scorecard. Scorecard information includes the scorecard name and unique identifier.

Strategy Map A strategy map is composed of perspectives, themes and objectives, and allows the linking of objectives in a cause-and-effect fashion. Strategy map information includes linkage cause, linkage effect, degree strength, interaction, and unique identifier for each link.

Target The level of performance or rate of improvement required for a particular measure. Targets are stated in specific units and should include time-based segments. Target information includes target name, description, unique identifier, update frequency, an optional owner reference, target value, and actual value.

Theme A descriptive statement representing a major component of a strategy, as articulated at the highest level in the vision. Themes represent vertically linked groupings of objectives across several scorecard perspectives. Theme information includes the theme name, description, unique identifier, an optional owner reference, and associated objectives.

4.6. Measurement Specification Template of the PSM

The Practical Software and Systems Measurement (PSM): A Foundation for Objective Project Management document [101] is a guide that describes how to define and implement a measurement program to support the information needs of software and system acquirer and supplier organizations. It is written for both government and industry organizations responsible for acquiring, developing, or maintaining software and systems. The guide addresses four major activities in the measurement process:

- Tailoring the software measures to address specific project issues.
- Applying software measures to convert the measurement data into usable information.
- Implementing a measurement process.
- Evaluating a measurement program.

PSM provides a *measurement specification template* [102] which defines the general data and implementation requirements for each measure. We briefly describe the most pertinent elements of the template.

Information Need Description	
Information Need	What the measurement user needs to know in order to make informed decisions.
Information Category	A logical grouping of information needs to provide structure for the information model.

Entities and Attributes	
Relevant Entities	The object that is to be measured. Entities include process or product elements of a project.
Attributes	The property or characteristic of an entity that is quantified to obtain a base measure.

Base Measure Specification	
Base Measures	A base measure is a measure of a single attribute defined by a specified measurement method.
Measurement Methods	The logical sequence of operations that define the counting rule to calculate each base measure.
Type of Method	The type of method used to quantify an attribute, either (1) subjective, involving human judgement, or (2) objective, using only established rules to determine numerical values.
Scale	The ordered set of values or categories that are used in the base measure.
Type of Scale	The type of the relationship between values on the scale, either: nominal, ordinal, interval, or ratio.
Unit of Measurement	The standardized quantitative amount that will be counted to derive the value of the base measure, such as an hour or a line of code.

Derived Measure Specification	
Derived Measure	A measure that is derived as a function of two or more base measures.
Measurement Function	The formula that is used to calculate the derived measure.

Indicator Specification	
Indicator Description and Sample	A display of one or more measures (base and derived) to support the user in deriving information for analysis and decision making. An indicator is often displayed as a graph or chart.
Analysis Model	A process that applies decision criteria to define the behaviour responses to the quantitative results of indicators.
Decision Criteria	A defined set of actions that will be taken in response to achieved quantitative values of the model.
Indicator Interpretation	A description of how the sample indicator was interpreted.

4.7. Software Measurement Metamodel of the FMESP Framework

García et al. [45] present a proposal for the integrated management of software measurement, which is part of the Framework for the Modeling and Evaluation of Software

Processes (FMESP) [44]. The proposal, which adopts the Model Driven Engineering (MDE) philosophy, provides a generic measurement metamodel to represent the meta-data related to the measurement process, a method to measure any kind of software entity represented by its corresponding metamodel, and GenMETRIC, a software tool that supports the framework.

Three entities are identified as candidates for measurement in the context of software development: process models which represent the different elements related to a process, projects which are concrete enactments of process models, and products which are obtained from carrying out projects.

The proposed *software measurement metamodel* is based on the concepts and relationships of an ontology for software measurement that the authors had developed previously. Figure 4.3 shows the UML diagram which shows the main elements of the software measurement metamodel. It is organized around four main packages which we describe briefly.

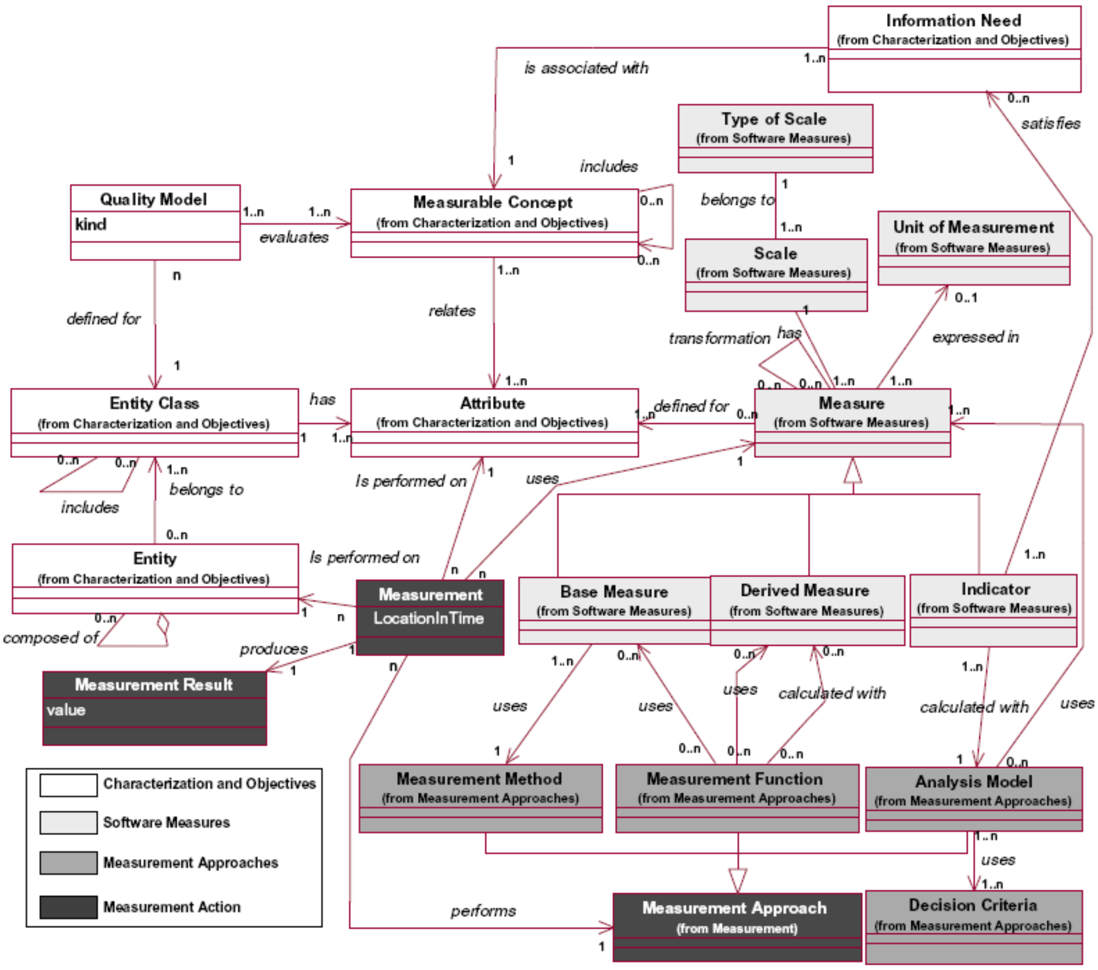


Figure 4.3.: Software measurement metamodel [45]

Software Measurement Characterization and Objectives includes the concepts required to establish the scope and objectives of the software measurement process.

The main goal of a software measurement process is to satisfy certain information needs by identifying the entities (which belong to an entity class) and the attributes of these entities (which are the object of the measurement process).

Software Measures aims at establishing and clarifying the key elements in the definition of a software measure. A measure relates a defined measurement approach and a measurement scale, and is expressed in a unit of measurement. Three kinds of measures are defined: base measures, derived measures, and indicators.

Measurement Approaches generalises the different approaches used by the three kinds of measures. A base measure applies a measurement method, a derived measure uses a measurement function (which rests upon other base and/or derived measures), and an indicator uses an analysis model (based on a decision criteria).

Measurement Action establishes the terminology related to the act of measuring software. A measurement is a set of measurement results, for a given attribute of an entity, using a measurement approach. Measurement results are obtained as the result of performing measurements.

The application of the proposed measurement metamodel consists in the definition of a specific measurement model—an instance of the measurement metamodel—that fulfils the organisation's information needs. Similarly, domain metamodels, that represent the candidate entities for measurement, are instantiated with domain models. Examples of domain models include UML models (use cases, class diagrams, etc.) of software applications and database models. Finally, the measures can be evaluated by GenMETRIC, a tool for the definition, calculation and visualisation of software measures.

Chapter 5.

Requirements on a Measure and Performance Indicator Model

In the preceding three chapters, we study the fundamental aspects related to measurement and performance measurement, and examine the few models for measures and performance indicators that have been proposed in literature. According to us, most of these models present a number of shortcomings. Some of the proposed models are very summary. They describe the most important concepts in a very brief manner, and do not specify the relationships that do exist between these concepts. Others provide sufficient detail, but do not consider all aspects that are relevant in the fields of measurement and performance measurement.

In order to be able to propose a model for measures and performance indicators, the requirements on the model need first to be established. Thus, the aim of the present chapter is to answer the following question: what are the requirements on a generic measure and performance indicator model?

We first discuss the concept of requirement and its importance in the context of information system development. We then list the requirements that we can derive from literature on measurement and performance measurement. They are grouped into three main categories: requirements which relate to measures, to performance indicators, and to strategy and performance.

5.1. Requirements and Requirements Elicitation

From a generic point of view, the term *requirement* can be considered as ‘a thing that is needed or wanted’ [95]. In information system development, requirements play an important role. All too often projects fail, exceed deadlines or budgets, or deliver solutions that do not meet the real needs of the users. Such failures and shortcomings are often related to the incomplete or incorrect discovery, understanding and/or management of requirements. In this particular context, Kotonya and Sommerville [68] define a requirement as ‘a statement of a system service or constraint’. Requirements define the services expected from the system and the constraints that the system must obey. The former are better known as *functional requirements*, they specify the functionality of the system. The latter are also called *non-functional requirements*. They place restrictions on the system been developed, notably in the fields of security, usability, reliability and performance.

Requirements engineering is 'the systematic process of eliciting, understanding, analysing and documenting these requirements' [68]. Requirements elicitation, also called requirements discovery, is probably the most critical step in the requirements engineering process. Kavakli and Loucopoulos [66] consider it as the 'understanding [of] the organizational situation that the system under consideration aims to improve, and describing the needs and constraints concerning the system under development'.

Different approaches exist to the elicitation of requirements. Interviews with stakeholders, for example, are a commonly used technique of requirements discovery. In a scenario-based approach, the interactions between a user and the system are described through scenarios. These scenarios can then be used to elicit and clarify system requirements. The goal-based approach considers the goals the system under consideration should achieve. Through a process of refinement and abstraction, goals are identified and linked to requirements that 'implement' the goals.

For a more detailed discussion on approaches to requirements elicitation, we refer the reader to [66, 114].

5.2. The Requirements

In this section, we list the requirements on a generic measure and performance indicator model. The requirements derive from literature in the fields of measurement and performance measurement, which we discuss in the preceding chapters.

Our approach to requirement elicitation could be considered as 'literature-based', since it relies on a broad review of domain-specific literature. Most, if not all of the requirements that we list can be considered as functional requirements, they describe services and functionalities that a system should be able to provide. These services and functionalities may or may not have an impact on the underlying model.

The requirements are grouped into three main categories: requirements which relate to measures, to performance indicators, and to strategy and performance.

5.2.1. Requirements related to Measures

R-1.1. Measurement is the process of assigning numbers to the properties of entities in a way that their original characteristics remain preserved.

R-1.2. A measure is a standard of measurement that allows to represent a property of an entity in a quantitative manner.

R-1.3. A measurement is either fundamental or derived [105, 111]. In fundamental measurement, an observed or empirical system is mapped to a numerical system which preserves all the relations and operations of the initial system. In derived measurement, new measures are defined in terms of existing ones.

R-1.4. In derived measurement, new measures are defined in terms of existing fundamental and/or derived measures using summary operators such as sum, average, ratio, percentage or rate [21, 61, 105, 111].

R-1.5. A measurement is relative to a specific scale [99, 105, 111].

R-1.6. A scale maps an empirical relational system to a numerical relational system by the means of a homomorphism (mapping function) [99, 105, 111].

R-1.7. A (regular) scale belongs to a scale type, the most common ones being absolute, ratio, interval, ordinal and nominal scale [99, 105, 111]. (In the case of derived measurement, no generally accepted theory exists [105]. Suppes and Zinnes [111] for example consider two scale types for derived measurements, one in the narrow sense, and another in the wide sense. For the sake of simplicity, we consider that every scale belongs to a single scale type.)

R-1.8. A scale type refers to a class of admissible transformations which specifies the transformations producing another homomorphism. [99, 105, 111].

R-1.9. The value of a measurement is expressed in a unit of measurement [25, 45, 76, 101].

R-1.10. Various criteria exist to evaluate the quality of measurement, notably accuracy, resolution, timeliness, reliability, and validity [20, 21, 61]. Accuracy expresses how well a measured value agrees with the real or standard value. Resolution indicates the smallest change that can be detected, and timeliness captures how often measurement values changes. Reliability refers to the consistency of a number of measurements, whereas validity refers to the extent to which a measure reflects the real meaning of the concept under consideration.

R-1.11. Measures (and performance indicators) can source data from integrated, but also from operational data sources [50, 69].

R-1.12. Measures (and performance indicators) can source data from internal, but also from external data sources [15, 23, 55, 115].

5.2.2. Requirements related to Performance Indicators

R-2.1. A performance indicator is a strategic instrument which allows to evaluate performance against targets.

R-2.2. Performance indicators rely on a standardised specification that includes all relevant attributes [30, 76].

R-2.3. Performance indicators are based on an explicitly defined formula and source of data [6, 49, 91, 101].

R-2.4. Performance indicators are 'owned' by individuals or groups who are accountable for their outcome and responsible for developing their methodologies [6, 30, 76, 79, 91, 101].

R-2.5. Performance indicators have a scope which states the areas of business or parts of the organization that are concerned [76].

R-2.6. Performance indicators can be classified according to different criteria. Common classifications are:

- leading and lagging indicators [Fitzgerald et al.] [6],
- internal and external indicators [Keegan et al., Lynch and Cross] [6, 63],
- financial and non-financial indicators [Keegan et al.] [62].

R-2.7. Performance indicators are grouped according to specific subject areas or points of view [6, 62, 76, 101].

R-2.8. Performance indicators are associated to targets. A target represents a value an organisation seeks to achieve at a specific moment in time [27, 49, 51, 64].

R-2.9. A target can represent a planned value, a norm, an expected value, or a constraint [101].

R-2.10. Targets may be owned by individuals or groups [3].

R-2.11. Two approaches exist to target setting: the static and the dynamic approach [49]. The static approach fixes the target at a certain performance level which remains unchanged. The dynamic approach expresses the target as a rate of expected improvement.

R-2.12. Targets are stated in specific units and should include time-based segments [3].

R-2.13. Performance indicators can be associated to a rating system which helps qualify the gap between the actual and the target value [11, 30].

R-2.14. A rating system is composed of a set of value ranges, each associated to a particular letter grade, score name or colour [11].

R-2.15. Performance indicators can be related to one another [8, 64, 100, 112]. Typical relationships include correlation, causality and conflict. In a correlation relationship, both indicators tend to change in a similar way, whereas in a causality relationship, the change in one indicator causes the change in the another.

R-2.16. Performance indicators can be structured hierarchically, where an indicator of a given level synthesises the indicators of inferior levels [18], or where an indicator is said to have an effect on a superior-level or equal-level indicator [112].

R-2.17. A performance indicator supports a particular organisational level within an organisation, typically the strategic, tactical, or operational level [Lynch and Cross] [6, 18].

R-2.18. The aggregation of performance indicators may require the normalisation of their values [18, 76, 104, 112].

R-2.19. Notes and comments can be associated to performance indicators [91].

5.2.3. Requirements related to Strategy and Performance

R-3.1. Performance indicators derive from strategy and objectives [49, 51, 62, 74, 79, 88]. An objective is 'a concise statement articulating a specific component of what the strategy must achieve or of what is critical to its success' [3].

R-3.2. Objectives may be owned by individuals or groups [3].

R-3.3. Two types of objectives exist: quantitative and qualitative objectives. Quantitative objectives specify a target value and a unit of measurement, while qualitative objectives are described more loosely, without a specific target value [33].

R-3.4. Objectives can be classified into three categories: strategic, tactical and operational objectives [12].

R-3.5. Objectives can be grouped according to specific subject areas or points of view [3, 62].

R-3.6. Objectives can be related to one another in cause-and-effect relationships [32, 63, 88].

R-3.7. Objectives can be decomposed hierarchically [12, 33, 69].

R-3.8. Initiatives are programs developed to achieve objectives or to close the gap between the performance level and the targets [3, 6, 49, 63, 88, 89].

R-3.9. Initiatives are limited in time and have a responsible person [3, 116].

R-3.10. A one-to-one correspondence between initiatives and performance indicators does not exist in general. Rather, an initiative may have an impact on one or several objectives [3, 63, 116].

Chapter 6.

Measure and Performance Indicator Model

In this chapter, we answer our fifth and last research question: how could a model representing generic measures and performance indicators look like?
We begin with examining what exactly models are, and the kinds of models that do exist. We discuss the increasing importance of models in the context of information system development, and study different approaches to the class discovery problem. Our model proposal for generic measures and performance indicators is then presented. We discuss its purpose and describe its most important aspects. We explain our design choices and state the patterns that are use in the model. Finally, we list a number of rules which apply to the model.

6.1. Models and Model-driven Engineering

6.1.1. Abstraction and Models

Blaha and Rumbaugh [16] consider a *model* as 'an abstraction of an entity of the real world'[1]. In this sense, *abstraction* is 'the principle of ignoring those aspects of a subject that are not relevant to the current purpose in order to concentrate solely on those that are' [94]. Thus, a model deliberately focuses on some aspects of the entity in question, while ignoring those aspects that are not relevant. Omitting superfluous details makes the model easier to manipulate than the original entity and reduces thereby complexity. Models are being used in many of fields, for instance in economy, geography, biology, and statistics. In the context of information systems, models have always played an important role, since they allow the represent problems and solutions at a higher level of abstraction than the actual code. The importance of representing a system at different levels of abstraction has been recognised since long. The Merise approach for instance

[1] Translated by the author.

advocates the use of models at three distinct levels of abstraction—the conceptual, logical, and physical level—where each level describes specific aspects and concerns of the system.
Frankel [42] proposes a basic taxonomy of different kinds of models, shown in figure 6.1. A *business model*, also called *domain model*, describes aspects of the business,

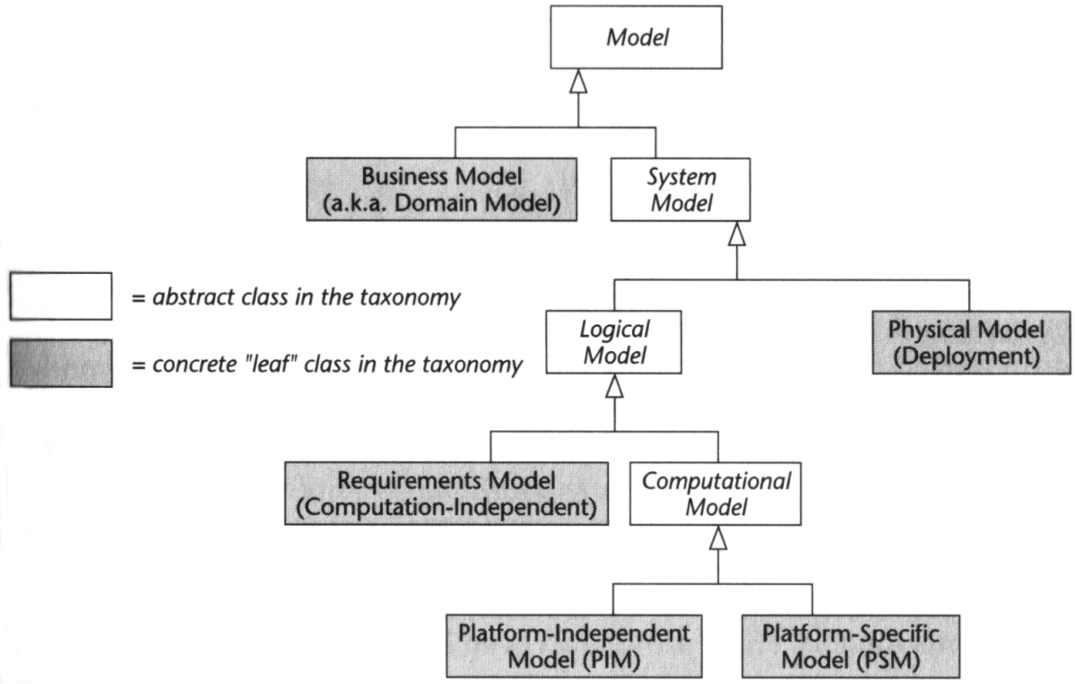


Figure 6.1.: Model taxonomy [42]

irrespective of whether those aspects are to be automated or not. Fowler (mentioned in [71]) gives the following definition: 'a domain model is a visual representation of conceptual classes or objects of the real world from a given domain'[2]. Mellor and Balcer [83] further specify what a domain is. 'A domain is an autonomous, real, hypothetical, or abstract world inhabited by a set of conceptual entities that behave according to characteristic rules and policies'.
A *system model* describes aspects of a system that automates certain elements of the business. Thus, the scope of a system model may be smaller than the scope of a corresponding business model. A *logical model* describes the logic of a system via structural and/or behaviour models. A *physical model* describes the physical artefacts and resources used during development and runtime. A *requirements model* describes the logical system in a computation-independent fashion, whereas a *computational model* describes the logical system, but takes technical factors into account. A *platform-independent model* is independent of any specific platform technology, meaning that it is independent of information-formatting technologies, programming languages and component middlewares. A *platform-specific model* on the other hand incorporates the aspects of particular platform technologies.

[2]Translated by the author.

6.1.2. Model-Driven Engineering and MDA

Mellor and Balcer [83] note that 'the history of software development is a history of raising the level of abstraction'. From machine code, over assembly languages, to third generation languages such as C++ and Java, the level of abstraction has constantly been raised, hiding thereby the details of the lower layers. The same authors further note that 'we formalize our knowledge of an application in as high a level language as we can'.
The *Model-Driven Engineering* (MDE) is probably the next higher layer of abstraction. MDE refers to the systematic use of models as primary artefacts throughout the engineering lifecycle. Thus, models are not simply considered as design artefacts that serve as guidelines and specification for programmers, but are a fundamental part of the whole production process, where the actual code is merely a consequence, a mechanical derivation of the models.

The *Model Driven Architecture* (MDA) [92] is the Object Management Group (OMG) proposal to the MDE paradigm. MDA is defined as 'an approach to IT system specification that separates the specification of functionality from the specification of the implementation of that functionality on a specific technology platform'. MDA proposes a methodical and architectural framework to the development and integration of systems that ensures the continuance of domain-specific aspects by decoupling them from technological preoccupations. Basically, the MDA approach consists in creating models and transforming them into other models based on the metamodels of the source and target model, and this from computation-independent models to platform-specific models.
Three types of models are distinguished in the MDA approach, each representing a different view of the system: the Computation Independent Model (CIM), the Platform Independent Model (PIM), and the Platform Specific Model (PSM). The CIM describes the requirements for the system and the environment in which the system will operate. The PIM describes the system at a high level of abstraction, but does not show details of the use of its platform. The PSM finally specifies how that system makes use of the chosen platform, it represents the implementation of a particular technology. Typical platforms include J2EE, .NET, and CORBA.
The model transformations and the traceability from CIM to PIM and PSM and vice versa form a key part of MDA. Model transformation is the process of converting one model to another model of the same system, as shown in figure 6.2. The input to a transformation is the source model and the specifications for transformation which are based on the mapping between the source metamodel and the target metamodel. Transformations can be done manually, with computer assistance, or automatically.
For a detailed discussion on MDA and its associated technologies, we refer the reader to [17, 42, 60].

Bézivin [24] argues that the MDE approach may lead to a paradigm shift in the field of software engineering, from 'everything is an object' to 'everything is a model'. Blanc [17] states three advantages expected from these new approaches: the continuance of domain-specific knowledge independent from technical aspects, a gain of productivity through automatic transformations of models, and the integration of platform-specific

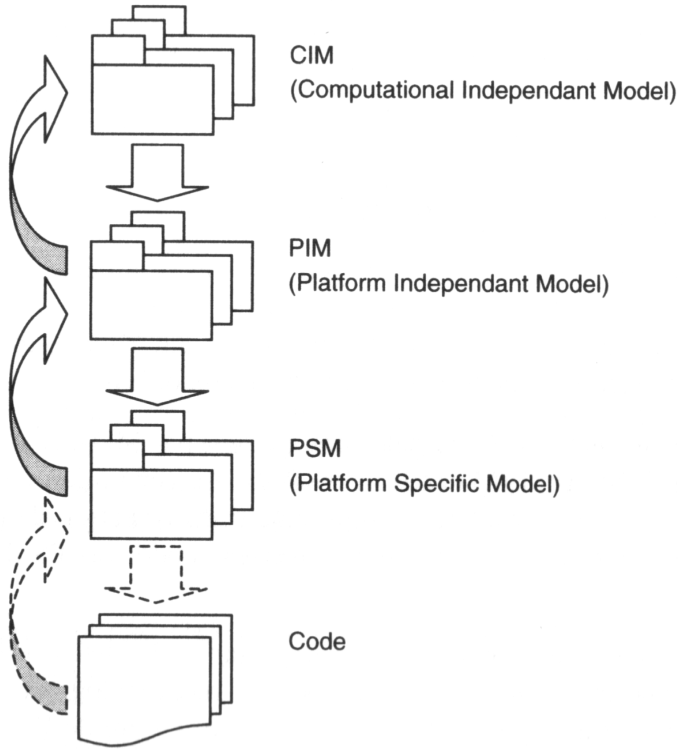


Figure 6.2.: General view of the MDA approach [17]

aspects into the models. However, MDE still faces many challenges. According to Schmidt [108], further research is needed, notably to support roundtrip engineering and synchronization between models and source code or other model representations. Furthermore, metamodelling environments and model interchange formats need to be standardised, and the specification and synthesis of model transformations need to be automated to simplify the evolution of models and metamodels.

6.1.3. Class Discovery Techniques

An object-oriented system decomposes its structure into *classes*, where a class represents an abstraction of real-world objects. The notion of class is central, since it is present from analysis to design, implementation and testing. Thus, finding a set of domain classes is a critical step in the development of object-oriented systems. Domain classes capture fundamental activities at the analysis level.
Different approaches exist for identifying classes from problem and requirement statements. According to Maciaszek [78] and Song et al. [110], the four most popular approaches are the noun phrase approach, the common class pattern approach, the use case driven approach, and the Class-Responsibility-Collaborators (CRC) approach.
The *noun phrase approach* advises to look for noun phrases in the requirements document. Every noun is considered a candidate class. The list of candidate classes is then divided into three groups: relevant, fuzzy, and irrelevant classes. Relevant classes are

those that manifestly belong to the problem domain, irrelevant classes are those that are outside of the problem domain. Fuzzy classes are those that can not confidently and unanimously be classified as relevant or irrelevant, they need further analysis.
The *common class pattern approach* derives candidate classes from the generic classification theory of objects. Classification theory is a part of science concerned with partitioning the world of objects into useful groups so that we can reason about them better. Bahrami (mentioned in [78]) lists the following groups for finding candidate classes: concept classes, event classes, organisation classes, people classes, and places classes. Other classification scheme do exit, Song et al. [110] list some of the class categories proposed in literature.
The *use case driven approach* is an approach emphasised by UML. A use case describes a functionality that a system provides through the interaction with an actor. It is usually composed of a textual description and different scenarios which describe the interactions between the actor and the system. These descriptions and scenarios are then used to discover candidate classes.
The *CRC approach* involves brainstorming sessions during which specially prepared cards are used. Each card corresponds to a class and has three compartments: one for the name, one for the responsibilities which represent the services the class provides, and one for the collaborators which are other classes required to fulfil the class' responsibilities. While executing a processing scenario, the participants fill the cards with the class names and assign responsibilities and collaborators. In this approach, classes are identified from the analysis of messages passing between objects.

6.2. The Model

In this section, we propose a model for generic measures and performance indicators. It has been elaborated based on the requirements identified in the preceding chapter. Furthermore, we provide a set of constraints that apply to this model.
In section 6.1.1, we see that models can be developed at different levels of abstraction. We situate our model at the domain (or conceptual) level, where a domain model can be considered as 'a visual representation of conceptual classes or objects of the real world from a given domain'[3] [71]. Thus, our model represents the concepts and objects relevant to the measurement and performance measurement domain.
The purpose of our model is twofold. Firstly, it aims at providing a better understanding of the concepts involved in this particular domain, as well as the relationships that exist between these concepts. Secondly, our model can act as a foundation to the development of a performance measurement system. Following the MDE approach described in section 6.1.2, our model could be transformed into a CIM model, which in turn could be transformed into a PIM, PSM, and finally into a concrete performance measurement system.
We do not pretend that our proposal represents *the* model for measures and performance indicators. Maciaszek [78] is right when saying that 'no two analysts will come

[3] Translated by the author.

up with the identical class models for the same application domain, and no two analysts will use the same thinking process when discovering the classes'. Thus, our model is nothing more than a proposal on how measures and performance indicators could be modelled. Specific measures and performance indicators may indeed require a different modelling.

6.2.1. Model Proposal

Our model proposal for generic measures and performance indicators is shown in figure 6.3. It is expressed using the class diagram notation of the Unified Modeling Language (UML). UML is, according to Rumbaugh et al. (mentioned in [78]), 'a general-purpose visual modelling language that is used to specify, visualize, construct, and document the artifacts of a software system'. A class diagram, one of the many diagrams proposed by UML, captures the static structure of a domain or system by characterising its objects. A class describes 'a group of objects having identical properties, a common behaviour, similar relations with other objects and a common semantic'[4] [16]. An object—which is an instance of a class—is 'a concept, an abstraction or an individually identifiable element having a sense for an application'[5] [16]. Thus, our model represents the static structure of the measurement and performance measurement domain, where a class describes a group of similar objects which are related in one way or another to these domains.
During the modelling process, three major aspects emerged around which the different classes could be organised: the measure aspect, the performance indicator aspect, and the aspect involving the organisation and its objectives. We briefly discuss these three aspects, the involved classes, and give examples.

Measure Aspect

Classes which are related to the measure aspect are shown on the left of the model. They are organised around the `Measure` class. A `Measure` allows to represent the `Property` of an `Entity` in a quantitative manner. For example, the entity object 'customer service' may have, amongst others, a property called 'average response time to complaints'. The measure object 'monthly average response time to complaints in department d' would then allow to quantify this property for a particular month and department. The value of a measure is expressed on a `Scale`. The value of the measure object 'operational cost per hour', for example, could be expressed on a Euro scale. A `Scale`, which maps one relational system into another, belongs to a particular `ScaleType`, the most common ones being the absolute, ratio, interval, ordinal, and nominal scale. Measures may be related to one another. A `Correlation` between two measures indicates to what extend the two measures vary together or oppositely. In the case of an

[4] Translated by the author.
[5] Translated by the author.

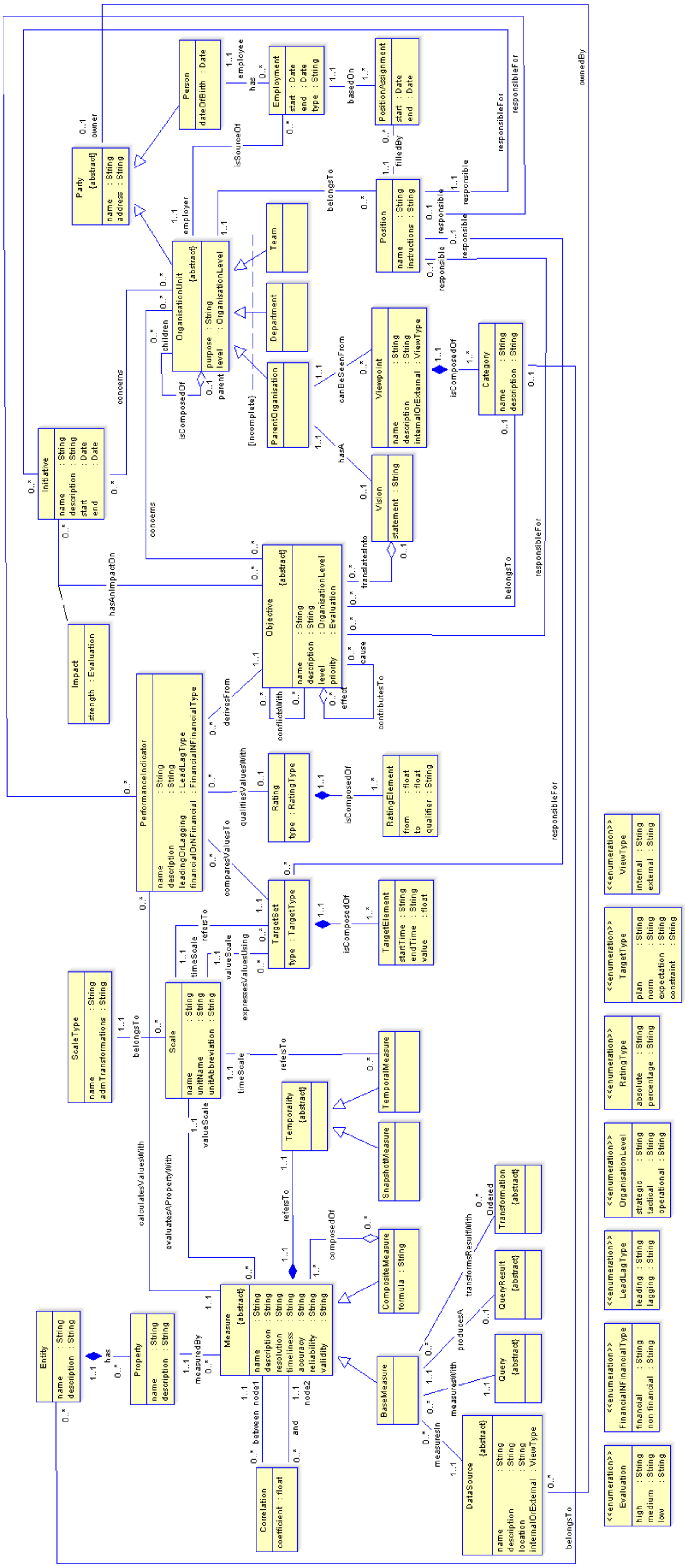


Figure 6.3.: Measure and Performance Indicator Model

e-commerce for example, the measure objects 'revenue' and 'stickiness'[6] may be correlated positively, meaning that both behave in a similar way.
A `Measure` is either a `BaseMeasure` or a `CompositeMeasure`. A `BaseMeasure` produces its value based on a particular `DataSource`, for example a database, data warehouse, or web service. A `Query` specifies the way that the required data can be accessed from the `DataSource`. `Transformations` define actions which need to be applied to the `QueryResult`, such as aggregation, formatting and rounding. A `CompositeMeasure` on the other hand combines the values of different base and/or composite measures, the particular combination being expressed in a `formula` using arithmetic and/or logical operators. The two base measure objects 'total expenses' and 'total revenue' can be used, for example, by the composite measure object 'efficiency ratio', which divides the former base measure by the latter.
From a temporal point of view, we distinguish between `SnapshotMeasures` and `TemporalMeasures`. A `SnapshotMeasure` does not take temporal aspects into consideration. It represents the value of a property at a particular time instance. The measure object 'current number of visitors' is a typical example. A `TemporalMeasure` on the other hand refers explicitly to a point/period in time, specified by a time `Scale`. The temporal measure object 'monthly number of visitors', for example, is relative to the month scale.
The quality of the value produced by a `Measure` can be evaluated through its `resolution`, `timeliness`, `accuracy`, `reliability`, and `validity`.

Performance Indicator Aspect

Classes related to the performance indicator aspect are represented in the middle of the model. They are organised around the `PerformanceIndicator` class. A `PerformanceIndicator` is a strategic instrument which allows to evaluate performance against targets. It can be classified as being either a leading or lagging indicator, and either a financial or non financial indicator.
An `Objective` is a statement of what is critical to success, and it is from these statements that performance indicators are derived. The performance indicator object 'customer satisfaction index', for example, could be an indicator for the objective object 'increase customer satisfaction'.
A `PerformanceIndicator` evaluates the actual level of performance through a `Measure`. The value of the measure may then be compared to the corresponding `TargetElement` value. `TargetElements` specify values for particular time periods and are organised into a `TargetSet`, which defines whether the values of its elements represent planned or expected values, or norms or constraints. Similar to a `Measure`, a `TargetSet` refers to a value `Scale` and a time `Scale`. For example, the performance indicator object 'average daily round-trip delay' for a given network could be associated to a target set object which stipulates that the values of its elements represent norms expressed on a millisecond scale, and that it is based on a day time

[6]Stickiness characterises the attractiveness of a site, section or page, and is measured by the average number of page views, session duration and page depth [8].

scale. A target element object of this set could indicate a value of 100 milliseconds between 01 January and 28 February, and another target element object 50 milliseconds between 01 March and 31 March.
A `Rating` allows to qualify a particular level of performance. It is either based on absolute values, or on percentages of the targeted value. A `Rating` is composed of `RatingElements`, which specify value/percentage ranges and corresponding qualifiers. In the case of a rating based on absolute values, values between 10 and 15, for example, may be considered as good, whereas those between 5 and 9 are only sufficient. In the case of a percentage-based rating, values between 80% and 100% of the targeted value may be considered as good, whereas those between 50% and 79% are sufficient, and so on.

Objective and Organisation Aspect

Finally, the classes which relate to the objective and organisation aspect are shown on the right of the model. They organise around the `Objective` and `OrganisationUnit` classes.
An `OrganisationUnit` represents a unit within the organisation's hierarchy, which may be composed of lower-level `OrganisationUnits`. A unit refers to a particular `OrganisationLevel`, either the strategic, tactical, or operational level. Each unit employs a number of `Persons` who occupy `Positions`. A `Position` may be responsible for particular `Objectives`, `TargetSets`, `PerformanceIndicators` and/or `Initiatives`. In our model, we specify three different units—`ParentOrganisation`, `Department`, and `Team`—where `ParentOrganisation` represents the top-level of the hierarchy. These units act as simple examples and may vary from one organisation to another.
A `ParentOrganisation` has a `Vision`, which represents a long-term goal of the organisation. A `Vision` can be translated into a number of strategic `Objectives`, which, in turn, may be decomposed into lower-level `Objectives`. For example, the objective object 'increase on-time deliveries' may contribute to the higher-level objective object 'increase customer satisfaction'. An `Objective` can also be in conflict with another `Objective`. The objective object 'reduce production costs', for example, may be in conflict with the objective object 'increase product quality'. An `Objective` refers to a particular `OrganisationLevel` and may be of more or less high `priority`. The achievement of an objective may be supported by `Initiatives`, which have an `Impact` on particular `Objectives`. Both `Objectives` and `Initiatives` may concern different `OrganisationUnits` within an organisation.
An organisation can be viewed from different `Viewpoints`, for example, from a financial, customer, internal processes, and learning and growth viewpoint. A `Viewpoint` is either internal or external, and may be decomposed into `Categorys`. The viewpoint object 'internal processes', for example, may be decomposed into the category objects 'process efficiency' and 'product quality'. A particular `Objective` belongs to one or the other `Category`.

6.2.2. Class Descriptions

In this section, we briefly describe each class of our model, their attributes and associations. The classes are ordered alphabetically based on the class name.

BaseMeasure
A measure which evaluates the property of an entity based on a particular data source.

dataSource [1] : DataSource	The data source which underlies the measure.
query [1] : Query	The query used to extract data from the data source.
queryResult [0..1] : QueryResult	The result produced by the query applied to the data source.
transformations [0..*] ordered : Transformation	The transformation(s) which are applied on the query result.

Category
A particular aspect of a viewpoint (e.g. process efficiency, customer feedback, personnel).

name : String	The name of the category.
description : String	A description of the category.
viewpoint [1] : Viewpoint	The viewpoint to which the category belongs to.
objectives [0..*] : Objective	The objective(s) that belong to the category.
entities [0..*] : Entity	The entity(ies) that belong to the category.

CompositeMeasure
A measure which combines a number of base and/or composite measures.

measures [1..*] : Measure	The measure(s) used by the composite measure.
formula : String	An expression which combines base and/or composite measures and/or constants using arithmetic and/or logical operators.

Correlation
A correlation expresses the degree to which two entities vary together or oppositely. The correlation coefficient may take any value between +1 and -1. The sign of the correlation coefficient defines the direction of the relationship, whereas the value of the coefficient measures the strength of the relationship.

node1 [1] : Measure	The first node of the relationship.
node2 [1] : Measure	The second node of the relationship.
coefficient : float	The coefficient of the correlation.

DataSource (abstract)
An abstract source of data.

name : String	The name of the data source.
description : String	A description of the data source.
location : String	The location of the data source.
internalOrExternal : ViewType	Specifies whether the data source is internal or external to the organisation.
owner [0..1] : Party	The owner of the data source.
baseMeasures [0..*] : BaseMeasure	The base measure(s) which are using the data source.

Department
A unit within an organisation.

Employment
A relationship between a person and an organisation.

employee [1] : Person	The person employed.
employer [1] : OrganisationUnit	The employer organisation.
start : Date	The start of the employment.
end : Date	The end of the employment.
type : String	The type of employment (e.g. full-time, part-time).
positionAssignments [1..*] : PositionAssignment	The position assignment(s) which are based on the employment.

Entity
A concept, object, process or any other element which can be subject to measurement (e.g. delivery, product, sales).

name : String	The name of the entity.
description : String	A description of the entity.
category [0..1] : Category	A generic category to which the entity belongs to.
properties [0..*] : Property	The properties that the entity has.

Evaluation (enumeration)
An enumeration of three simple terms—high, medium, low—which help qualify concepts such as strength and priority.

FinancialNFinancialType (enumeration)
An enumeration of two performance indicator types: financial or non financial indicator.

Impact
The impact that an initiative has on an objective.

objective [1] : Objective	The objective on which the initiative has an impact.
initiative [1] : Initiative	The initiative which as an impact on the objective.
strength [1] : Evaluation	The strength of the impact of the initiative on the objective.

Initiative
A program developed to achieve objectives and close performance gaps.

name : String	The name of the initiative.
description : String	A description of the initiative.
start : Date	The data at which the initiative starts.
end : Date	The data at which the initiative ends.
organisationUnits [0..*] : OrganisationUnit	The units which are concerned by the initiative.
responsible [1] : Position	The position within an organisation which is responsible for the initiative.
objectives [0..*] : Objective	The objective(s) on which the initiative has an impact.

LeadLagType (enumeration)
An enumeration of two performance indicator types: leading or lagging indicator.

Measure (abstract)
A standard of measurement that represents a property of an entity in a quantitative manner.

name : String	The name of the measure.
description : String	A description of the measure.
resolution : String	The smallest change that the measure can reflect.
timeliness : String	It indicates how often the value of the measure changes.
accuracy : String	It expresses how well the value of the measure agrees with the real or standard value.
reliability : String	It indicates the consistency of a number of measurements.
validity : String	It expresses the extent to which the measure reflects the real meaning of the property under consideration.
property [1] : Property	The property of an entity that the measure represents.
temporality [1] : Temporality	The temporal aspect of the measure.
valueScale [1] : Scale	The scale used to express the value of the property.
compositeMeasures [0..*] : CompositeMeasure	The composite measure(s) that use the measure.
performanceIndicators [0..*] : PerformanceIndicator	The performance indicator(s) that are based on the measure.
correlations [0..*] : Correlation	The correlation relationship(s) in which the measure is taking part.

Objective
A statement of what is critical to success. Objectives can be structured either hierarchically or as graphs in a cause-and-effect manner.

name : String	The name of the objective.
description : String	A description of the objective.
level [1] : OrganisationLevel	The organisational level at which the objective can be situated.
priority [1] : Evaluation	The priority of the objective.
vision [0..1] : Vision	The vision which the objective translates.
category [0..1] : Category	The category to which the objective belongs.
responsible [0..1] : Position	The position within an organisation responsible for the objective.
organisationUnits [0..*] : OrganisationUnit	The unit(s) which are concerned by the objective.
initiatives [0..*] : Initiative	The initiative(s) which have an impact on the objective.
causes [0..*] : Objective	The objective(s) which contribute to the achievement of the objective.
effects [0..*] : Objective	The objective(s) on which the objective has an effect.
conflicts [0..*] : Objective	The objective(s) with which the objective is in conflict.
performanceIndicator [0..*] : PerformanceIndicator	The performance indicator(s) which derive from the objective.

OrganisationLevel (enumeration)
An enumeration of the levels within an organisation's structure: strategic, tactical, and operational level.

OrganisationUnit (abstract)
An abstract unit of an organisation.

purpose : String	The purpose of the organisational unit.
organisationLevel [1] : OrganisationLevel	The organisational level at which the unit is situated.
parent [0..1] : OrganisationUnit	The parent unit.
children [0..*] : OrganisationUnit	The child unit(s).
positions [0..*] : Position	The position(s) that exist in the unit.
employments [0..*] : Employment	The employment(s) concluded between the unit and a person.
objectives [0..*] : Objective	The objective(s) which concern the unit.
initiatives [0..*] : Initiative	The initiative(s) which concern the unit.

ParentOrganisation
The top-level organisational unit which englobes all units.

vision [0..1] : Vision	The vision of the organisation.
viewpoints [0..*] : Viewpoint	The point(s) of view from which the organisation can be seen.

Party (abstract)
An abstract entity that describes both persons and organisations.

name : String	The name of the party.
address : String	The address of the party.
dataSources [0..*] : DataSource	The data sources the party owns.

PerformanceIndicator
An instrument which derives from an objective and which allows to measure performance against targets.

name : String	The name of the performance indicator.
description : String	A description of the performance indicators.
leadingOrLagging [1] : LeadLagType	Indicates whether the performance indicator is a leading or lagging indicator.
financialOrNFinancial [1] : FinancialNFinancialType	Indicates whether the performance indicator is a financial or non financial indicator.
measure [1] : Measure	The measure which allows to evaluate the actual performance level.
targetSet [1] : TargetSet	The targets which are set for the performance indicator.
rating [0..1] : Rating	The rating which allows to qualify a level of performance.
objective [1] : Objective	The objective from which the performance indicator derives.
responsible [0..1] : Position	The position within an organisation which is responsible for the performance indicator.

Person
A human being.

dateOfBirth : Date	The date of birth of the person.
employments [0..*] : Employment	The employment(s) concluded by the person.

Position
A function within an organisation (e.g. manager, accountant, salesman).

name : String	The name of the position.
organisationUnit [1] : OrganisationUnit	The organisation to which the position belongs to.
positionAssignments [0..*] : PositionAssignment	The assignment(s) to the position (current and past).
instructions : String	The work instructions associated to the position.
objectives [0..*] : Objective	The objective(s) for which the position is responsible for.
performanceIndicators [0..*] : PerformanceIndicator	The performance indicator(s) for which the position is responsible for.
targetSets [0..*] : TargetSet	The target(s) for which the position is responsible for.
initiatives [0..*] : Initiative	The initiative(s) for which the position is responsible for.

PositionAssignment
The assignment of a person employed by an organisation to a position.

employment [1] : Employment	The employment on with the assignment is based.
position [1] : Position	The assigned position.
start : Date	The start of the assignment to a position.
end : Date	The end of the assignment to a position.

Property
A characteristic of an entity (e.g. percent of on-time deliveries, software size, sales growth).

name : String	The name of the property.
description : String	A description of the property.
entity [1] : Entity	The entity to which the property refers to.
measures [0..*] : Measure	The measure(s) that allow to evaluate the property.

Query (abstract)
An abstract representation of a query which allows to access data from a data source.

baseMeasures [0..*] : BaseMeasure	The base measure(s) which are using the query.

QueryResult (abstract)
An abstract representation of the result produced by a query on a data source.

baseMeasure [1] : BaseMeasure	The base measure which produced the result.

Rating
A rating allows to qualify a particular level of performance.

type [1] : RatingType	The type of rating, absolute or percentage. Absolute rating specifies absolute value ranges, whereas percentage rating specifies percentage ranges based on the targeted value.
ratingElement [1..*] : RatingElement	The rating element(s) which specify ranges and corresponding qualifiers.
performanceIndicator [0..*] : PerformanceIndicator	The performance indicator(s) which use the rating.

RatingElement
An element which specifies a value/percentage range and a corresponding qualifier.

from : float	The start of the value/percentage range.
to : float	The end of the value/percentage range.
qualifier : String	A score name, letter grade, mark, or colour which qualifies values situated in the value range (e.g. good, bad, A, B, 6, 1, green, red).
rating [1] : Rating	The rating to which the rating element belongs to.

RatingType (enumeration)
An enumeration of two types of rating: absolute and percentage rating. Absolute rating specifies absolute value ranges, whereas percentage rating specifies percentage ranges based on the targeted value.

Scale
A scale allows the mapping of an (empirical) relational system to a numerical relational system. ‘Kilogram’, ‘Euro’, ‘centigrade’, and ‘month’ are typical scale examples.

name : String	The name of the scale.
unitName : String	The name of the scale unit.
unitAbbreviation [0..1] : String	An abbreviation used for a scale unit.
scaleType [1] : ScaleType	The scale type to which the scale belongs to.
measures [0..*] : Measure	The measure(s) which use the scale.
targetSets [0..*] : TargetSet	The target(s) which use the scale.

ScaleType
A scale type specifies the transformations that are admissible when changing from one scale to (another of the same type). Common scale types include absolute, ratio, interval, ordinal, and (nominal scale.

name : String	The name of the scale type.
admTransformations : String	The admissible transformations of the scale type.
scales [0..*] : Scale	The scale(s) which are of this type.

SnapshotMeasure
A measure which does not take temporal aspects into consideration. It represents the value of a property at a particular time instance (e.g. current number of visitors, current number of units on stock).

TargetElement
An element which specifies a targeted value for a specific point/period in time.

startTime : String	The point in time from which on the target value is valid (depends on the time scale).
endTime : String	The point in time at which the validity of the target value ends (depends on the time scale).
value : float	The value of the target element.
targetSet [1] : TargetSet	The target set to which the target element belongs.

TargetSet
A set of values which are planned, expected, or which represent norms or constraints.

type [1] : TargetType	The type of target.
timeScale [1] : Scale	The time scale to which the target elements refer to.
valueScale [1] : Scale	The scale used to express a target value.
targetElement [1..*] : TargetElement	The target element(s) which express time-dependant target values.
performanceIndicator [0..*] : PerformanceIndicator	The performance indicator(s) which refer to the target set.
responsible [0..1] : Position	The position within an organisation responsible for the target set.

TargetType (enumeration)
An enumeration of possible target types: plan, norm, expectation, and constraint.

Team
A unit within an organisation.

Temporality (abstract)
A temporality is an abstract representation of the temporal aspect of a measure.

measure [1] : Measure	The measure to which the temporality belongs.

TemporalMeasure
A measure which refers explicitly to a point/period in time (e.g. monthly revenue, weekly percentage of on-time deliveries).

timeScale [1] : Scale	The time scale to which the measure refers to (e.g. year, quarter, month, week, day).

Transformation (abstract)
An abstract representation of a transformation which can be applied on the result of a query. Typical transformations include aggregation, formatting and rounding.

baseMeasure [0..*] : BaseMeasre	The base measure(s) that use the transformation.

Viewpoint
The point of view from which an organisation can be seen (e.g. customer, internal processes, financial, learning and growth).

name : String	The name of the viewpoint.
description : String	A description of the viewpoint.
internalOrExternal [1] : ViewType	Specifies whether the viewpoint is internal or external to the organisation.
parentOrganisation [1] : ParentOrganisation	The organisation to which the viewpoint belongs to.
categories [1..*] : Category	The different categories that exist within the viewpoint.

ViewType (enumeration)
An enumeration of the two types of views: internal or external view.

Vision
A long-term goal of an organisation.

statement : String	A statement that outlines the vision of the organisation some years into the future.
parentOrganisation [1] : ParentOrganisation	The organisation to which the vision belongs.
objectives [0..*] : Objective	The strategic objective(s) into which the vision translates.

6.2.3. Design Choices and Patterns

Design Choices

During the model design process, we have been confronted several times with situations where we had to choose amongst a number of design alternatives. We briefly describe these situations and explain our choices.

Measurement theory differentiates between fundamental and derived measurement [105, 111]. In fundamental measurement, an observed or empirical system is mapped to a numerical system, whereas in derived measurement, new measures are defined in terms of existing ones. In the context of IT-supported measurement systems, the use of these two terms may not be adequate, since we may only have limited knowledge of the way the underlying data has been produced. A particular data element may already be the result of divers derivations and transformations, making a distinction between fundamental and derived measurement difficult to establish.
Therefore, we do not use the terms 'fundamental' and 'derived', but 'base' and 'composite'. A base measure quantifies the property of an entity based on a particular data source, whereas a composite measure is the combination of two or more base measures and/or composite measures and/or constants using arithmetic and/or logical operators.

Some concepts, such as the concept of measure, can be viewed from different angles.

If we consider the way a measure produces its value, we can distinguish between base measures and composite measures. Measures can also be considered from the point of view of their temporality. Some measures, we call them temporal measures, refer explicitly to a point or period in time. The measure 'monthly sales revenue' is an example. Other measures, we call them snapshot measures, do not take temporal aspects into consideration. They represent the state of a property of an entity at the point in time at which the measurement takes place. For instance, the measures 'current number of visitors' and 'current number of units on stock' make no reference to time and depend on the point in time at which the measure is calculated.
Modelling particular aspects of concepts usually involves the use of inheritance, where elements are organised according to their similarities and differences. In order to avoid a situation of multiple inheritance in the modelling of the above aspects, we use an approach called *delegation* (described in Blaha and Rumbaugh [16]), where some aspects of an object are transferred to a related object. In our model, the processing aspects of a measure are represented through inheritance from an abstract measure super-class. The temporal aspects of a measure are delegated to the abstract temporality class, which is linked to the measure class through a composition association. Thus, any combination of processing and temporal aspects is possible without having to rely on multiple inheritance.

What exactly is a performance indicator, and how do you represent it in a model? Is it a measure, or more precisely a specialisation of a measure? Is it an independent class which is associated to other classes such as measure, target and objective? Or is a performance indicator merely a concept which designates a particular constellation of objects?
We have long been thinking about these questions, and in previous versions of our model, one or the other approach has been emphasised. Finally, we have decided to represent a performance indicator as an independent concept. Thus, a performance indicator is not a measure (a specialisation of a measure), but is associated to a measure. Basically, a performance indicator associates the concepts which are fundamental in performance measurement, which are objectives, measures, and targets. This approach allows a better separation of concerns. Measures simply produce measurement values and are not directly concerned with objectives and targets. They can be reused in the definition of new measures and may provide values for different performance indicators (for example with a different objective and targets). Objectives on the other side are not directly linked to targets and measures, which provides greater flexibility when modelling a strategy.

Several authors propose approaches on how objects such as measures, performance indicators, and objectives should be structured and the kind of relationship they can entertain with objects of the same class (described in sections 3.3.2 and 3.5.2). Some authors propose hierarchical structures, while others propose graphs. As to the relationships, the most often cited ones are causality, correlation, and conflict. Some the proposals have been included in our model, while others have been discarded. For instance, we do not consider conflict relationships in the context of measures and performance indicators. Objectives may indeed be in conflict with others. For example, the objectives 'increase

quality' and 'reduce cost' may be in conflict. But can a measure or an indicator be in conflict with another? We do not think so.
In our model, objectives can be structured as hierarchies or as graphs using causality relationships. For example, the objective 'increase on-time deliveries' may have an effect on the objective 'increase customer satisfaction'. Furthermore, conflict relationships which may exist between objectives can be represented.
As to performance indicators, we consider that their structure implicitly depends on the structure of the related objectives. Having two parallel but related structures—an objective structure and a performance indicator structure—may indeed lead to inconsistencies and would therefore require a bigger control and synchronisation effort.
As to the relationships between measures or between performance indicators, we find only correlation to be pertinent since it is statistically verifiable. A correlation expresses the degree to which two entities vary together or oppositely. Thus, correlation analysis allows to determine measures (or performance indicators) whose values vary together (both increase/decrease together) or oppositely (as one increases, the other decreases and vice versa). In our model, correlation can be specified for measures.

Patterns

A *pattern* is, according to Larman [71], 'a named description of a problem and a solution which may be applied to new contexts'[7]. Patterns formalise and generalise established knowledge, idioms and proven principles in order to make them reusable in different contexts. In addition to the description of the problem and the solution, a pattern usually provides an example, some advice on how to use it, its strengths and weaknesses, and possible variations.
Patterns can be found at all phases of development. Eriksson and Penker [33] consider three types of patterns depending on the problem domain they address: business, architectural, and design patterns. *Business patterns* address problems within a business domain, typically in the analysis phase. *Architectural patterns* address problems related to the architectural design of information systems, such as the organisation of systems and subsystems. *Design patterns* finally focus on technical solutions that are flexible and adaptable. Other classifications for patterns do exist. Gamma et al. [43] for instance define the following categories: creational, structural, and behavioural patterns. *Creational patterns* concern the process of object creation. *Structural patterns* deal with the composition of classes and objects, whereas *behavioural patterns* characterise the ways in which classes and objects interact and distribute responsibility.
We briefly describe the patterns that are used in our model. They fall in the business and design pattern categories and deal with structural issues.

Organisation and Party Pattern The Organisation and Party pattern, described by Eriksson and Penker [33], allows a flexible representation of an organisation's struc-

[7]Translated by the author.

ture which can easily be adapted in case of changes. We use the basic structure of this pattern in our model, represented by the classes `OrganisationUnit`, `ParentOrganisation`, `Department`, and `Team`. This basic structure explicitly states the different organisational units. Although the extended structure of this pattern is much more flexible (the different types of organisations are not stated explicitly), we use the basic structure since it allows us to show several associations that only particular organisational units entertain.

Employment Pattern The Employment pattern, described by Eriksson and Penker [33], structures the relationship between a person and an employer. It specifies factors such as the contract of employment between a person and an organisation, the position(s) and responsibilities that are assigned to a person, and start and end dates of the employment and of held positions. The employment pattern is represented in our model by the classes `Party`, `Person`, `OrganisationUnit`, `Employment`, `Position`, `PositionAssignment`, and their corresponding associations.

Composite Pattern The Composite Pattern, described by Gamma et al. [43], allows to compose objects into tree structures to represent component-composite structures. The pattern is composed of a component, leaf, composite and client class. We use this pattern in our model to represent composite measures which as composed of other base and/or composite measures. The `Measure` represents the component, the `BaseMeasure` represents the leaf, the `CompositeMeasure` represents the composite, and the `PerformanceIndicator` is the client.

6.2.4. Rules and Constraints

Rules are an essential part in any organisation. Eriksson and Penker [33] define a business rule as ‘a statement that defines or constraints some aspects of the business, and represents business knowledge’. Thus, rules regulate how an organisation operates and is structured. They ensure that an organisation is run in conformance with external (e.g. laws and regulations) and internal restrictions.
Some rules may be enforced directly by the information system that supports the organisation’s activities. Thus, it is important that rules are formalised in order to complement the models. Eriksson and Penker [33] consider three categories of business rules: derivation rules, constraint rules, and existence rules. *Derivation rules* define how information in one form may be transformed into another form, they link information together and show dependencies. *Constraint rules* specify the possible structure or the behaviour of objects or processes. They can be divided into three subcategories: structural, operational/behaviour, and stimulus/response rules. *Structural rules* specify conditions on the static structure that must always hold. *Operational/behaviour rules* define pre- and postconditions that constrain what must be true before or after an opera-

tion is performed. *Stimulus/response rules* finally specify that certain actions should be performed when certain events are generated. The last category, *existence rules*, govern when a specific object may exist.
While some rules may be embedded directly in a model (the multiplicity of an association in a UML class diagram for example forms a structural constraint), others may require an explicit formulation. We list a number of rules here that derive from the requirements stated in the preceding chapter, or that we can assume based on common sense. Most of these rules fall in the constraint rules category and define structural aspects.

C-1. A measure does not correlate with itself.

C-2. The measures involved in a composite measure must not form any circular dependencies.

C-3. A set of targets associated to a performance indicator must use the same value scale as the measure associated that performance indicator.

C-4. A set of targets associated to a performance indicator which in turn is associated to a temporal measure must use the same time scale as that temporal measure.

C-5. The starting point in time of a target element must be before its ending.

C-6. The time range of a target element must not overlap with the time ranges of other target elements of the same set.

C-7. The value/percentage range of a rating element must not overlap with the value/percentage ranges of other rating elements of the same rating.

C-8. A strategic objective can not be a sub-objective of a tactical or operational objective.

C-9. A tactical objective can not be a sub-objective of an operational objective.

C-10. An objective can not have itself as sub- or upper-objective.

C-11. An objective can not be in conflict with itself.

C-12. Only strategic objectives having no upper-objective can be associated to a vision.

C-13. An organisation unit situated at a strategic level can not be a sub-organisation unit of an organisation unit situated at a tactical or operational level.

C-14. An organisation unit situated at a tactical level can not be a sub-organisation unit of an organisation unit situated at an operational level.

C-15. Circular dependencies between organisation units are not allowed.

C-16. An organisation unit can not have itself as sub- or upper-organisation unit.

C-17. A person employed by an organisation unit holds a position within this same organisation unit.

C-18. The starting date of an employment must be before its ending.

C-19. The starting date of a position assignment must be before its ending.

C-20. The starting date of a position assignment must not be before the starting date of the related employment.

C-21. The ending date of a position assignment must not be after the ending date of the related employment.

Chapter 7.

Performance Measurement System Prototype Application

In this chapter, we present a prototype application of a performance measurement system. The purpose of this application is to evaluate the measure and performance indicator model that we have proposed in chapter 6.
First, a generic architecture for performance measurement systems is proposed, based on which the application will be build. We then present our prototype application. We describe different aspects relative to the application, and present the use cases that it translates. Technical aspects of the implementation are discussed, and a simple scenario is presented which illustrates the basic functioning of the application. Finally, we evaluate our model based on the findings we made during the development phase of the prototype application.

7.1. Architecture Proposal

In this section, we propose a high-level architecture for performance measurement systems based on which our prototype application will be implemented. The proposed architecture is very generic, since we do not target any particular usage domain. It is based on a distributed three-tier architecture, comprised of a client-, middle-, and resource-tier. We did choose this kind of architecture based on a study of measurement systems that have been proposed in literature (see appendix C). However, depending on the purpose of a particular performance measurement system and the environment in which it is to operate, a number of specific requirements may exists, some of which may have a direct impact on the design of the architecture. Thus, the architecture that we propose here is nothing more than a general purpose architecture which may be suitable for some cases, but not for others.

7.1.1. Overview

'Software architecture seeks to organize complex systems by separating concerns', writes Frankel [42]. The *separation of concerns* is indeed important since it makes a system much more flexibility than monolithic systems when it comes to changing certain aspects of a system. Thus, the separation of concerns tends to 'localise' changes to a certain aspect of a system. Furthermore, it promotes the reuse of logic and data.
Multi-tiered architecture is a widely accepted approach for distributed information systems which emphasises the separation of concerns. Our architecture is based on a classic three-tier architecture, composed of a client-, middle-, and resource-tier. Figure 7.1 shows the overall architecture.

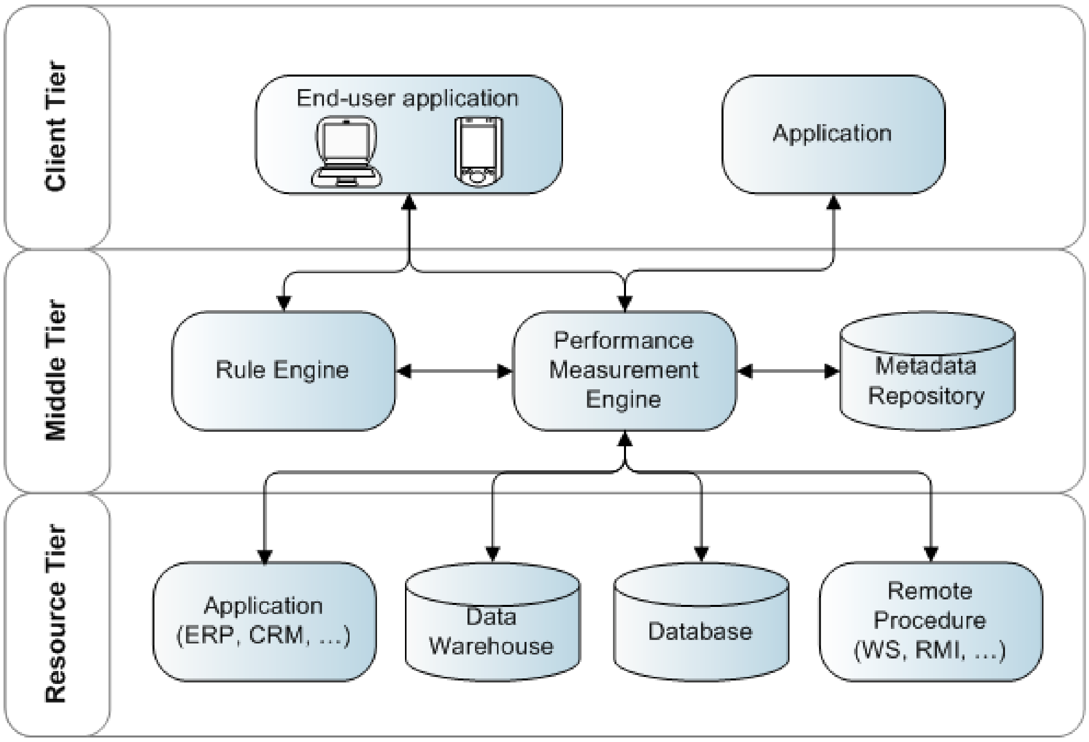


Figure 7.1.: Architecture proposal for a performance measurement system

We use the term *client-tier* instead of 'presentation-tier' or 'front-end' since this tier may be composed of any kind of application, be it an application which presents information to end-users, or an application that requires information for its own processing needs.
The *middle-tier*, also called 'business-tier' or 'enterprise-tier', encapsulates all the logic of the performance measurement system. The performance measurement engine handles all the tasks related to performance measurement, the required metadata being stored in a metadata repository. An optional rule engine allows to handle rules associated to the performance measurement process.
Finally, the *resource-tier*, also called 'database-tier' or 'back-end', is composed of any system capable of providing data for measurements.

7.1.2. Client-Tier

The *client-tier* is composed of any kind of application which, for one reason or another, requires performance information. Typical end-user applications of performance measurement systems include dashboards and analysis tools. A *dashboard* is a visual display of the most important information, typically measures, performance indicators and other information relevant to the user's needs. Dashboards are usually desktop or web-based applications that are accessed on PCs or laptops, mobile wireless devices such as mobile phones and Personal Digital Assistants (PDAs) are other devices from which a user might wish to access performance information. Dashboards are discussed more in detail in the next subsection.

While performance information is mainly destined for end-users, other applications, which require this kind of information for processing purposes, may also be potential clients of a performance measurement system. For example, a process management system may require performance information on a particular work process in order to be able to fine-tune the parameters of related processes.

A performance measurement system may indeed have a number of heterogeneous clients. Thus, it is crucial that performance information is accessible over one or several standardised interfaces.

Dashboards

Just as the dashboard of a car provides critical information to operate the vehicle, a 'digital' dashboard provides the user with a summary of the most important information he or she needs to achieve his or her objectives. Few [38] proposes the following definition: a dashboard is 'a visual display of the most important information needed to achieve one or more objectives which fits entirely on a single computer screen so it can be monitored at a glance'. Figure 7.2 shows a typical example of a dashboard.

Dashboards have several essential characteristics that differentiate them from other tools [18, 37]. Dashboards provide a coherent vision of the objectives to achieve, they allow to compare, diagnostic, and control the actual level of performance with regards to these objectives, and indicate values that have fallen behind targets or passed over specific thresholds. Furthermore, they act as a reference in a team by presenting a common vision of the situation, and facilitate the communication and exchange between different actors.

By definition, dashboards are limited to a single screen to keep all the information within eye span. Therefore, information is condensed, primarily in the form of summaries and exceptions. Summaries are aggregated values and typically represent measures and performance indicators. Exceptions are values that have fallen out of normality and represent problems or opportunities, they manifest themselves as alerts and triggers. Small and concise display media are used to communicate the data and its message. The most common display media found on dashboards are bar and line graphs, pie charts, scatter plots and radar graphs.

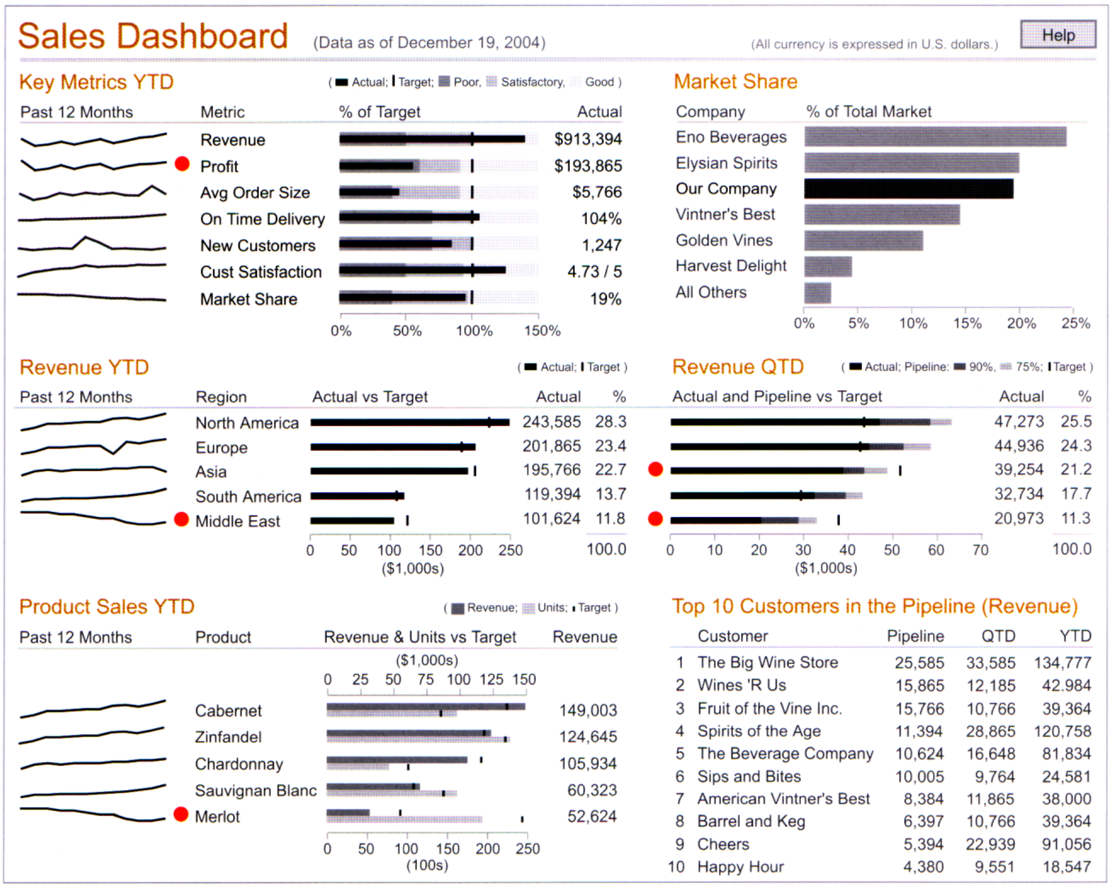


Figure 7.2.: A sample sales dashboard [38]

Dashboards are often used at different levels within an organisation. Three major types of dashboards are generally distinguished, depending on the organisational level which they support: strategic, analytical (or tactical), and operational dashboards [12, 31, 38]. Strategic dashboards reflect a global vision on the organisation. They usually focus on high-level performance indicators and may include forecasts and histories. Analytical (or tactical) dashboards indicate the performance of processes and products and are mid-term oriented. Finally, operational dashboards allow an immediate overview of operational tasks and processes and maintain awareness of events that might require attention or response.
For information on the visual design of dashboards, we refer the reader to Few [38].

7.1.3. Middle-Tier

The *middle-tier* holds all the logic of the performance measurement system, it acts as a layer between the client-tier that requests for performance information and the resource-tier that holds the raw data. Its main components are the performance measurement engine, the metadata repository, and the rule engine.

The *performance measurement engine* is responsible for the computing of measures and performance indicators and allows the management of all elements relevant to measurement and performance measurement. For instance, it allows the management of measure and performance indicator specifications, target values, ratings, and objectives. The performance measurement engine is accessed over a standardised interface, such as a web service. For example, when a client application requests for information on a particular performance indicator, the performance measurement engine fetches the required metadata from the metadata repository, sends queries to the data sources specified by the measure which is associated to the indicator, calculates the overall measurement value, determines the corresponding target value, rates the measurement value, and send the results back to the client.
The *metadata repository* holds all the data and metadata necessary for the performance measurement engine to operate. It stores metadata such as measure and performance indicator specifications, as well as data on targets, ratings, and objectives. The schema of the metadata repository derives from the model that we propose in chapter 6.
The *rule engine* is a component that monitors critical events and values, delivers alerts to users, and initiates system actions if necessary. For example, if the value of a particular measure deviates from a predefined threshold, the rule engine may notify interested people automatically via e-mail or SMS, or may trigger a particular system action. We discuss rule engines more in detail in the next subsection.

Rule Engines

Rule engines allow the monitoring of critical events and values. Contrary to the traditional on-demand mode where information is requested explicitly by a client, rule engines operate by themselves according to a set of rules. In the context of performance measurement, a user may for instance declare that he or she wants to be notified via SMS every time that the value of a particular performance indicator is qualified as insufficient. The rule engine then monitors constantly the given performance indicator and sends an SMS to the user every time that the value of the indicator is rated as insufficient.
A rule engine generally contains three distinct components: rules, actions and recipients.
The *rules component* manages the rules at work. A simple rule could for instance sound like this: the percentage of defective units produced is superior to 5. Every time that a rule is evaluated as true by the rules component, an alert is created. Each organisation has its own sets of rules to support its activities. Therefore, the rules component needs to be sufficient flexible as to accommodate any possible rule. Dresner [28] defines different rule types: static rules, duration rules, temporal rules, coincidence rules, frequency rules, and delta rules. Rules can also be combined with one another, resulting in complex rules that trigger alerts only if certain conditions are met.
The *actions component* defines the follow-up action when an alert occurs. A typical action could constitute of sending an e-mail to relevant people. E-mail action may be sufficient when the reaction to an alert requires human decision and when an immediate

inaction is not adverse. Otherwise, one or several system actions may be necessary. A system action involves the transmission of control commands with specified parameters to a system. Typical system actions include the updating of a database, and the launching or stopping of specific processes. For example, if an inventory level goes below a minimum threshold, an alert may trigger a purchase order in the system for a pre-set quantity. System actions should always be recorded in order to be able to trace actions that have been undertaken by the system.
The last component, the *recipients component*, determines the recipient(s) of a user alert. User alerts may be addressed to individuals and/or to user groups. For instance, the validation of a particular rule may trigger an action that consists of sending an e-mail or an SMS to a specific user group.
As an example, the Business Process Cockpit, a process monitoring system proposed by Sayal et al. [107], incorporates a rule engine which can execute three kinds of actions in response to an alert. It can notify a given user either via e-mail or SMS, or send a message to a Java Message Service (JMS) bus to be consumed by any client interested in receiving the event. It can also provide feedback to the process management system in order to modify system parameters or change the course of a process.

7.1.4. Resource-Tier

The *resource-tier* is composed of any system capable of providing data, be it locally or over a network. Transactional databases, data warehouses and data marts are probably the most frequently used data sources. Domain-specific applications such as Enterprise Resource Planning (ERP), Customer Relationship Management (CRM) and Supply Chain Management (SCM) systems may also provide valuable data for measurements. Remote Procedure Calls (RPCs) such as web services and Remote Method Invocations (RMIs) are other possible data sources. Another valuable source of data are system logs. Sayal et al. [107] for instance propose a system that monitors processes based on the data logged by a process management system. Similarly, Fenstermaker et al. [34] propose a system that measures the design process of semiconductors based on the design tool's log files.
The data sources of a performance measurement system may be internal or external to the organisation. Partner organisations, for example, may grant access to certain information, and third party data providers may deliver the kind of data which is unavailable in the organisation's data stores.
In the following subsection, we briefly discuss the problem of data integration, which is frequent in organisations maintaining multiple data stores.

Data Integration

Within an organisation, numerous heterogeneous systems and data sources do usually co-exist. Depending on the specific technology of a system, data is accessed in different ways. Most modern systems can be accessed via a standardised information exchange protocol such as the Open Database Connectivity (ODBC) or web services. But there are still systems that use proprietary file formats which makes the access to their data difficult. Furthermore, data elements are often stored in different formats, they may not have the same semantics and refer to different contexts, which makes them incomparable. Thus, obtaining a unified and coherent view of all the data poses numerous problems.

Data integration is 'the problem of combining data residing at different sources, and providing the user with a unified view of these data' [73]. Two approaches are generally considered to solve this kind of problem: a tightly coupled and a loosely coupled approach.

In a tightly coupled approach, data from several sources are extracted, transformed, and loaded (ETL) periodically into a single data repository, typically a data warehouse or a data mart. Data warehouses are data stores that hold data from operational data sources in an aggregated, summarised form. Data marts are similar applications, but are designed to serve only the specific needs of a particular group of users. The advantage of this approach is that data can be queried with a single schema. However, problems can arise with the 'freshness' of data, for example when an original data source is updated, but the warehouse still contains the older data.

A recent approach in data integration is to loosen the coupling between data. The idea is to provide a uniform query interface over a mediated schema of a virtual database and to consider the data sources as materialised views of the mediated schema. When the mediated schema is queried, the data integration solution transforms the query into specialised queries over the original data sources. Wrappers are adaptors for the individual data sources that execute the local queries and transform the results for the data integration solution. Finally, the results of the queries are combined into an answer to the user's query. A convenience of this solution is that new data sources can be added by simply constructing an adaptor for them. However, the mediated schema needs to be rewritten whenever a new data source is to be integrated or an existing source changes its schema. The commercial application of this approach is known under the name of Enterprise Information Integration (EII).

7.2. The Prototype Application

7.2.1. Description and Use Cases

In chapter 6, we propose a measure and performance indicator model. In order to be able to evaluate the model, at least on a small scale, we propose a prototype application of a performance measurement system which builds upon our model. The Software Engineering Institute (SEI) [109] defines a prototype as 'a preliminary type, form, or instance of a product or product component that serves as a model for later stages or for the final, complete version of the product'. Thus, our prototype application provides

only a very limited number of functionalities and features, and is far from being a full-fledged performance measurement system.
The prototype application builds upon a simplified version of our model which features only its core elements, which are the elements directly related to measures and performance indicators. Other elements, as those related to an organisation, person or viewpoint, have been discarded for the sake of simplicity. Furthermore, we consider only snapshot measures having no parameters. Those elements of our model that have been defined at an abstract level, such as a data source or a query, have been transformed into concrete elements. Appendix D shows the database schema which derives from this simplified model and which is used by the metadata repository to hold data and metadata.
From a technical point of view, our prototype application is based on the architecture proposal presented in section 7.1. It is composed of a client application that runs on a mobile phone, a performance measurement engine, a metadata repository, and a database system which holds the raw data for measurements. For the purpose of demonstration, we have implemented a set of rather common, unrelated measures and performance indicators from the fields of finance, manufacturing, and networking.

In order to be able to evaluate our model, we have defined five use cases, which represent basic functionalities that every performance measurement system should be able to perform, independent of the domain in which is being used. The use cases are the following:

1. monitor specific measures and performance indicators,
2. list all measures,
3. list all performance indicators,
4. display measure details,
5. display performance indicator details.

The first use case, 'monitor specific measures and performance indicators', allows a user to monitor a set of measures and performance indicators on a mobile phone. The information is displayed on a miniature dashboard and is updated automatically every few seconds as to keep the information 'fresh'. For performance indicators, the information being displayed includes the indicator's name, the measured value, the target value and the rating of the measured value. Measures are displayed with their names and the measured values.
The use cases 'list all measures' and 'list all performance indicators' allow to display the names of all measures and performance indicators that are available. Finally, the use cases 'display measure details' and 'display performance indicator details' allow the user to access detailed information on a particular measure or performance indicator. In the case of performance indicators, the indicator's name, description, type (leading or lagging indicator), point of view, objective, measured value, target value, and rating

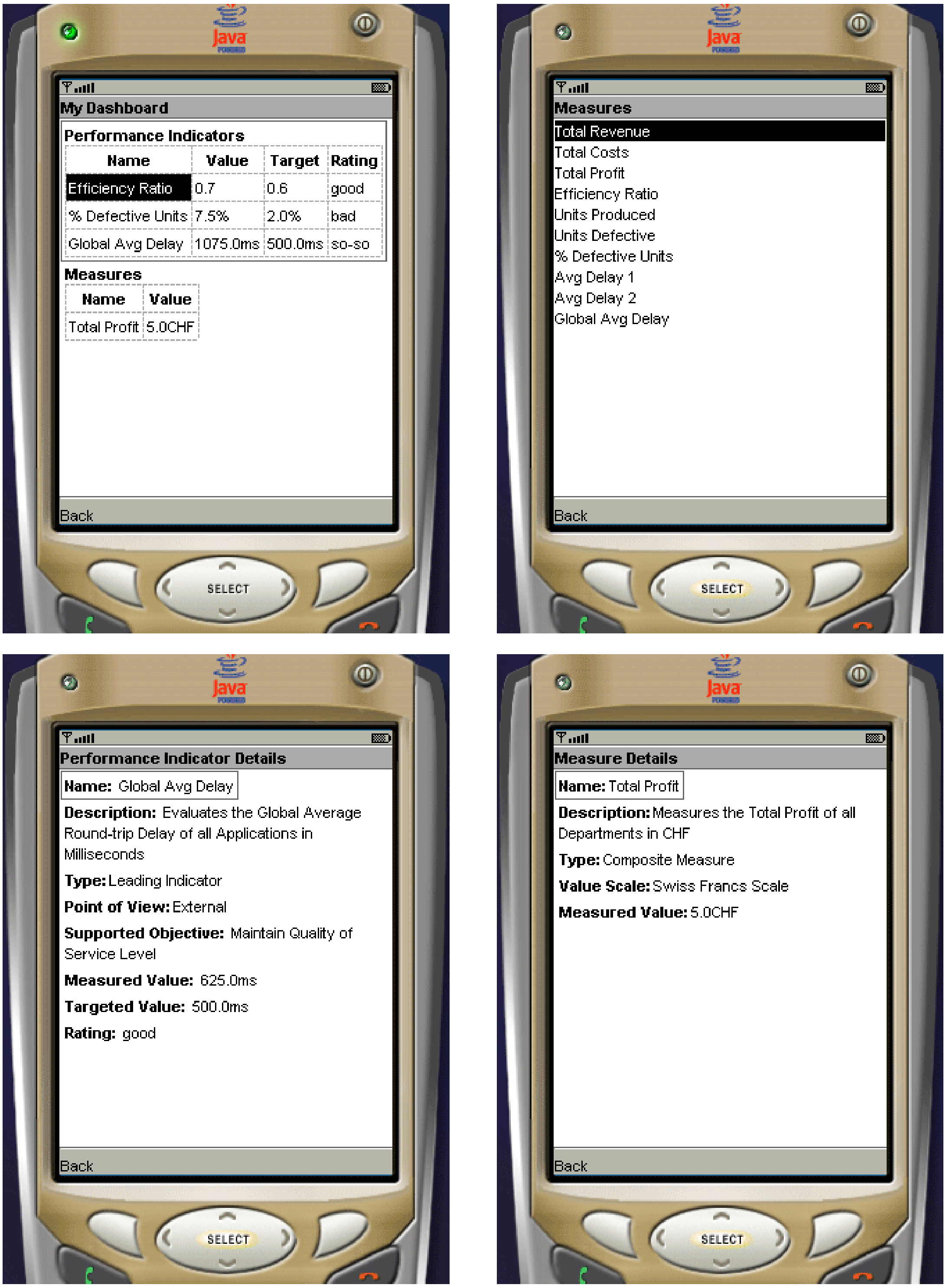


Figure 7.3.: Screenshots of the mobile phone client application

are displayed. In the case of measures, the measure's name, description, type (base or composite measure), value scale, and measured value are displayed.

Some screenshots of the mobile phone client application of our prototype are shown in figure 7.3. The first screenshot shows the dashboard, which translates our first use case. It displays information on three performance indicators—the financial efficiency ratio, the percentage of defective units produced, and the global average round-trip delay of communications on networks. The dashboard further displays the measured value of the total profit measure. The second screenshot shows a list of available measures, a similar display exists for performance indicators. They represent the second and third use case. The third screenshot shows a detailed view on a particular performance indicator, the average round-trip delay indicator, while the forth screenshot shows details on the total profit measure. They translate the forth and fifth use case.

7.2.2. Technical Aspects

Our prototype application is based on a distributed, three-tiered architecture, as lined out in section 7.1. The client-tier component runs on a mobile phone. The performance measurement engine of the middle-tier runs on a Java EE platform and uses a database system as metadata repository. Finally, the resource-tier is represented by a database system. A rule engine is not part of our prototype application, since it is out of the scope of this research. Figure 7.4 shows the overall architecture of the prototype application.

The application client runs on a Java Platform, Micro Edition (Java ME) which provides an environment for applications running on mobile devices. It allows to display a miniature dashboard which displays information on a set of measures and performance indicators. The screen is updated automatically every few seconds by a timer task. Furthermore, the application client allows to list all available measures and performance indicators, and to display detailed information on each one of them.
Since nearly all information displayed by the application client is provided by the middle-tier, it requests data whenever needed. For this purpose, the client calls the web service interface of the middle-tier component. Web services are 'web-based applications that use open, XML-based standards and transport protocols to exchange data with calling clients' [58]. The Simple Object Access Protocol (SOAP) is used to exchange data. SOAP defines the envelope structure, encoding rules, and conventions for representing web service invocations and responses. The requests and responses are transmitted over the Hypertext Transfer Protocol (HTTP). The WS Proxy, which represents the remote web service on the application client, is generated by the Java API for XML Web Services (JAX-WS), based on the Web Services Description Language (WSDL) file of the web service. Whenever the application client requires data, it simply invokes the methods on the WS Proxy.

The performance measurement engine runs on a Sun Java System Application Server Platform, a compliant implementation of the Java EE 5 platform. Its main component

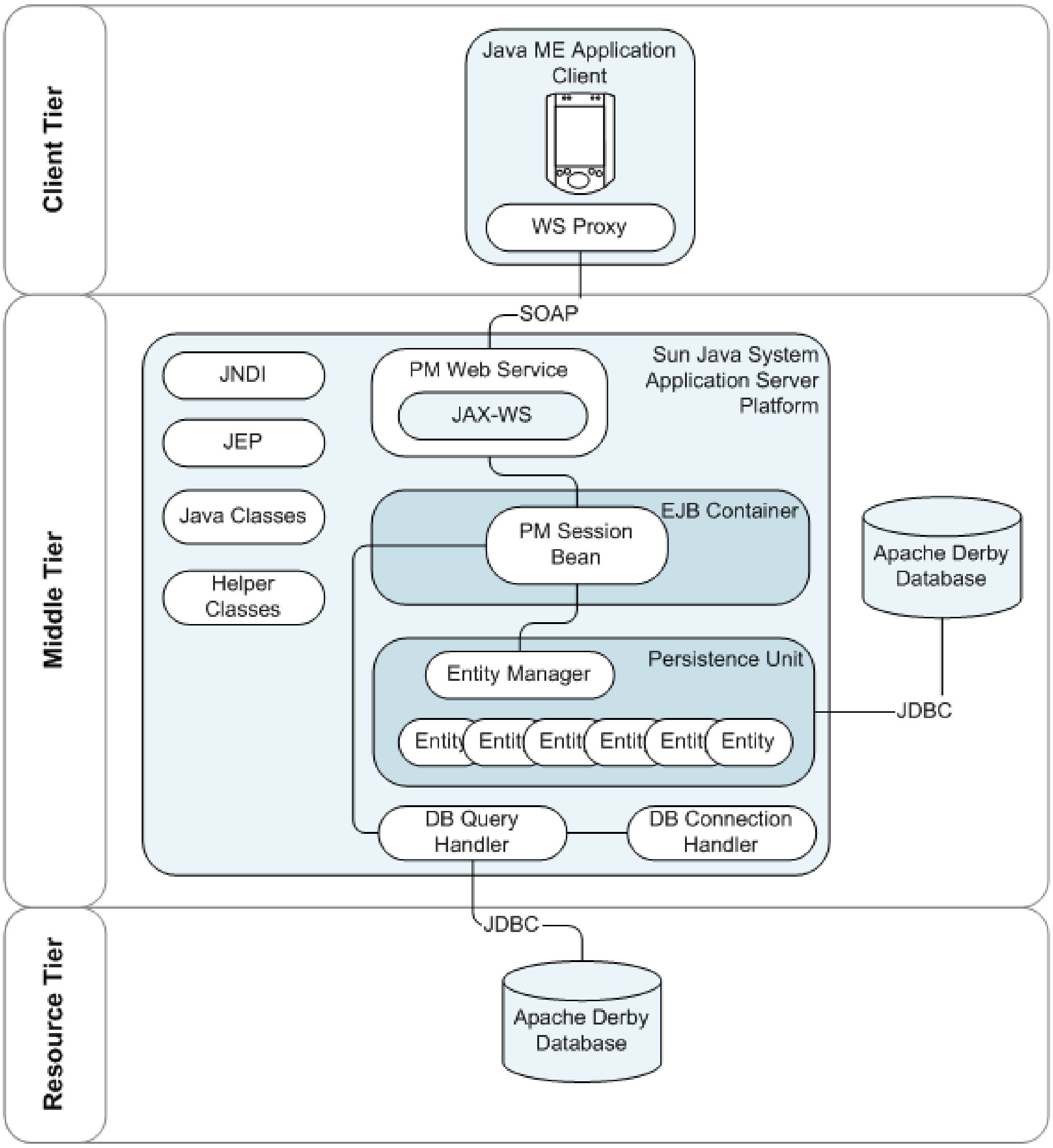


Figure 7.4.: Architecture of the prototype application

is the PM Session Bean, which is an Enterprise JavaBean (EJB) component. An EJB is ‘a server-side component that encapsulates the business logic of an application’ [58]. Different kinds of EJBs do exist, in our case, we use a stateless Session Bean. A stateless Session Bean represents a conversation with a client. Contrary to a stateful Session Bean, a stateless Session Bean does not maintain a conversational state with the client. EJBs run in an EJB container, a runtime environment which manages the execution of the components and acts as an interface between the components and the platform-specific functionalities.
To access the middle-tier component, a client invokes the Session Bean’s methods. In the case of our prototype, the application client can invoke several methods of the PM Session Bean which provide information on measures and performance indicators. The application client accesses the PM Session Bean through the Bean’s web service endpoint implementation class, which we call the PM Web Service, and which is also generated by the JAX-WS.
The metadata repository of the middle-tier is represented by an Apache Derby database, a relational database written entirely in Java. It stores all the metadata on measures and performance indicators, as well as data on targets, ratings and objectives. The mapping between the object-oriented component and the relational database is achieved through Java Persistence. An Entity represents a table in the relational database, and each Entity instance corresponds to a row in a table. The Persistence Unit defines a set of all Entity classes that are managed by an Entity Manager. This set of Entity classes represents the data contained in a data store, in our case, it is the data stored in the metadata repository. Entities are managed by an Entity Manager which is responsible for creating and removing Entity instances, finding Entities, and allowing queries to be run on Entities. The query language used, called the Java Persistence Query Language, uses an SQL-like syntax. The queries operate over an abstract schema of the Entities and are then translated into queries that are executed over that database schema to which the Entities are mapped. Thus, when the PM Session Bean requires data from the metadata repository, it asks the Entity Manager to execute a particular query. The query is operated over the abstract schema, translated into an SQL query, and executed on the database. The result of the query is then translated into the corresponding Entity instances. The connection between the Persistence Unit and the metadata repository is established through the Java Database Connectivity (JDBC), which provides methods for querying and updating data in a database.
In addition to the PM Session Bean, the performance measurement engine uses different Java classes in order to accomplish its tasks. The DB Connection Handler and the DB Query Handler for instance manage the execution of SQL queries addressed to the database of the resource-tier. The Java Naming and Directory Interface (JNDI) is used to locate different resources, such as the JDBC resource of the metadata repository. The Java Math Expression Parser (JEP) is a Java library for parsing and evaluating mathematical expressions, we use it to evaluate the formulas of composite measures.

Finally, the resource-tier is represented by an Apache Derby database which stores all the ‘raw’ data. It is on this database that the performance measurement engine executes the queries that are specified by the measures stored in the metadata repository.

7.2.3. Scenario

In order to illustrate the functioning of the prototype application, we describe a simple scenario where a user requests detailed information on a particular performance indicator. Although the actual prototype application works slightly differently, this example illustrates well the basic functioning of our application. Figure 7.5 shows a sequence diagram of the example scenario.

Our user, let us call him Pierre, is a manager in a big manufacturing company with its head office in Geneva. He is responsible for the new production site which opened five months ago in the outskirts of Shanghai, China. Pierre is currently on his way to work and sits in the bus. Lately, the team in China encountered problems with some of the machinery and the production was running behind schedule. Thus, he wants to check the current state of production since he has another five minutes to ride on the bus.
Pierre launches the client application on his mobile phone and asks for detailed information on the performance indicator 'Units Produced, Shanghai'. The performance measurement engine, which is able to produce this information, resides on a server of an affiliated service provider. The client application invokes the corresponding method of the WS Proxy, which represents the web service of the performance measurement engine on the client side, and which handles all the data exchanges. The WS Proxy generates a SOAP request, sends it over HTTP to the PM Web Service of the performance measurement engine, which finally addresses the request to the PM Session Bean.
In order to be able to answer the client's request, the PM Session Bean requires metadata, which is stored in the metadata repository. In our case, the PM Session Bean requires metadata on the performance indicator 'Units Produced, Shanghai', its associated measure which specifies how to calculate the measurement value, the targets which have been set, the rating which is to be used, as well as the objective the performance indicator supports. Thus, the PM Session Bean requests the Entity Manager to prepare the metadata. The Entity Manager retrieves the metadata from the metadata repository and instantiates the corresponding Entity classes.
In our scenario, the site in China disposes of two production lines, each storing relevant data in a separate database. Thus, the measure associated to our performance indicator, let us call it 'Number of Units Produced L1 + L2', is a composite measure, composed of the two base measures 'Number of Units Produced L1' and 'Number of Units Produced L2'. Each of these two base measures specifies the location of the database, the required authentication data, as well as the SQL query which allows to produce the measurement value.
To determine the value of each of these two base measures, the PM Session Bean sends the database information and the SQL query to the DB Query Handler. This handler requests the DB Connection Handler to create a connection to the database, executes the SQL query on that connection, and sends the measured value back to the PM Session Bean. The formula of the composite measure, which specifies how to calculate the overall value, is then evaluated. In our case, the values of the two base measures are simply added. The measures that we illustrate here are rather simple examples, more complex measures are of course possible.

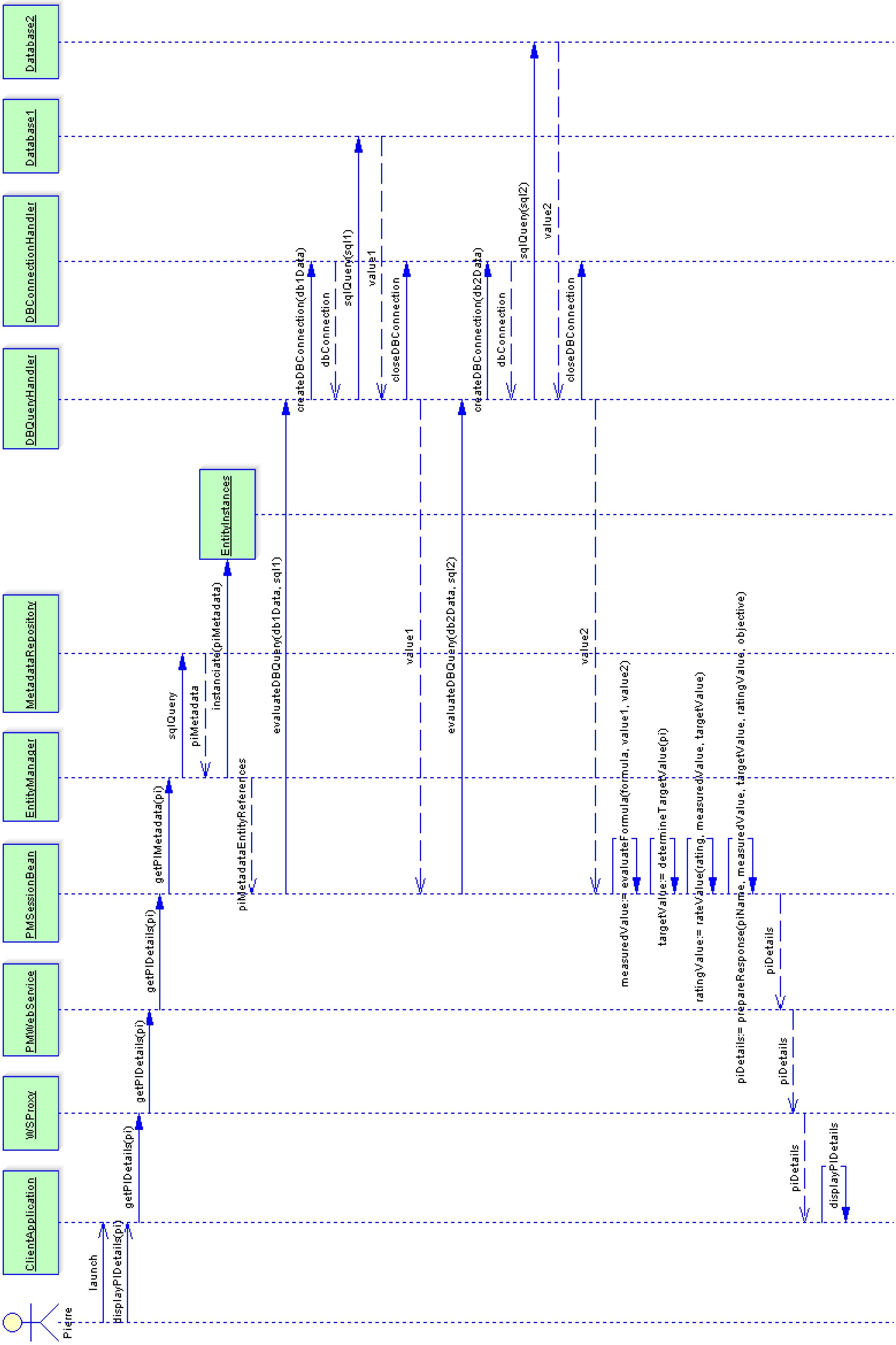


Figure 7.5.: Sequence diagram of an example scenario

Having established the measured value of the composite measure associated to the performance indicator, the PM Session Bean determines the value which is targeted by the performance indicator. It then qualifies the measured value based on the target value and the rating associate to the indicator. Finally, the PM Session Bean prepares the client response, which includes the name of the performance indicator, the measured value, the target value, the value of the rating, as well as the objective the indicator supports. It then sends the response back via SOAP to the mobile phone of Pierre.
It seems that the production in China is back on track. Although the number of produced units is slightly behind target, it is still qualifies as being good. Pierre is relieved. Having still a bit less than five minutes to ride on the bus, he starts reading one of the free newspapers lying around.

7.2.4. Evaluation

The purpose of our prototype application is to evaluate the measure and performance indicator model that we propose in chapter 6. Since the model is rather large in size, we have decided to concentrate on its core elements, which are those directly related to measures and performance indicators. Furthermore, we have considered only parameterless snapshot measures in the prototype application.
In order to be able to evaluate the model, we have defined five use cases, 'monitor specific measures and performance indicators', 'list all measures', 'list all performance indicators', 'display measure details', and 'display performance indicator details'. They represent basic functionalities that every performance measurement system should be able to perform. Thus, the measure and performance indicator model that we propose should be able to support them.

During the development phase of the application, we have encountered no particular problems related to the model. The metadata repository, whose schema derives from a simplified version of the model, was able to provide all the data required by the performance measurement engine to answer the client application's requests. No modifications to the model were necessary. Thus, we may say that the model that we propose allows to represent all the data necessary to translate the above stated use cases.
The way that measures are conceived in our model, as base measures or composite measures, has allowed us to define measures in a very flexible manner. Measures are considered as simple data providers. They can easily be reused for the definition of new, more complex measures, which, in turn, may be reused for still other measures. Thus, any combination of base and/or composite measures using arithmetic operators is possible. (The ability to use logical operators in the formulas of composite measures has not been implemented, to our knowledge, they are only rarely used in measurement definitions.)
In section 6.2.3, we wonder how performance indicators should be modelled, and finally argue that they ought to be represented as an independent concept. This way, a performance indicator is not a specialised measure, but associated to a measure. This clear separation of concerns has allowed us to handle the different elements of the model with

much more ease. Considering performance indicators as specialised measures, as it is the case in some already existing models, introduces according to us only unnecessary complexity.

From a technical point of view, we can say that the prototype application works as we have ‘imagined’ it. Apart from some minor issues, no major problems have been encountered during the development phase.
The application client has been tested on a mobile phone emulator only. Running this application on a real phone should not pose any particular problems, as long as it supports the Java ME platform. As to the data transmission between the application client and the performance measurement engine, the amount of data being exchanged is fairly limited. Furthermore, the prototype application follows the traditional consumer-producer paradigm, which requires a low uplink and a high downlink. Thus, the bandwidth provided by 2G or 2.5G mobile communication systems should be sufficient for the data exchange.
We would like to point out that a mobile phone client application is just one of the many potential clients of a performance measurement system. Such a system would typically have a web-based or desktop front-end which allows the monitoring of measures and performance indicators on a dashboard, as well as the management of all involved aspects. Moreover, applications that are not primarily targeted for end-users, but require performance information for their own processing purposes, are other potential clients.

Although the prototype application that we have presented here did validate a simplified version of our model, more thorough evaluation is necessary. Aspects that have not been taken into consideration in the prototype, such as temporal measures, parametrised measures, and aspects related to objectives, should be further evaluated. Moreover, the measures and performance indicators that we have specified, as well as the data that we have used for measurements, do not derive from a ‘realistic’ context, but have been defined for the sole purpose of demonstration. The use of our model in the context of another project, for example in the field of process management or networking, would provide a good opportunity to evaluate it on a larger scale, and under more realistic circumstances.

Chapter 8.

Conclusion and Outlook

The measurement of organisational performance has become a major issue in recent years, and this not only for profit-oriented, but also for public and non-profit organisations. The ability to measure the vital activities of an organisation, and this at all organisational levels, is indeed critical to its success in today's fast-paced world.
Performance measurement has been the subject of a lot of research since the 1990's, which led to numerous recommendations and frameworks. At the heart of every performance measurement process are measures and performance indicators. They allow to evaluate whether the organisation is on track, or whether actions need to be initiated that help achieving its objectives. Although central to the performance measurement process, few authors actually define what exactly a measure or performance indicator is, what the aspects are that need to be taken into consideration, and how they relate to each other. The few models that have been proposed in literature are not sufficient in our eyes, they either lack detail or do not consider all aspects relevant to the performance measurement process. In the present research, we have tried to solve this shortcoming.

In the chapter entitled *Measurement*, the foundation to measurement activities has been laid out. Although no consensus seems to exist between the different researchers having contributed to the measurement theory, we have considered fundamental and derived measurement as the main approaches to measurement. We have seen that scales play a central role, for it is them that specify how one relational system is mapped into another.
In the *Performance Measurement* chapter, the findings from an extensive literature review on performance measurement have been presented. We have identified the concepts that are of importance in this domain, and have discussed them from different points of view. Due to the large number of frameworks that have been proposed in literature, we have restricted our discussion to a single one—the balanced scorecard framework—which is probably the most popular approach to performance measurement. At this point, it has become clear to us that representing all the different proposals that have been made in literature in a single model would not be possible. Although not contrary, many authors pursue approaches that are not necessarily compatible with others.
In *Existing Measure and Performance Indicator Models*, we have described seven models that have been proposed in literature. All of them could qualify as 'generic', although

some are targeted for specific domains. While some of these models are very summary and lack detail, others do not consider all aspects that are relevant in measurement and performance measurement.
Based on our findings in the fields of measurement and performance measurement, we have presented the *Requirements on a Measure and Performance Indicator Model*. They summarise the different aspects and approaches discussed in literature, and which need to be taken into consideration when designing measures and performance indicators.
We have finally proposed a *Measure and Performance Indicator Model*. It summarises the structural aspects relative to measures, performance indicators and the performance measurement process in general, in a single, comprehensive model. The model highlights the concepts that are of importance, and shows how these concepts relate to each other. Thus, it allows a better understand of the structural aspects involved in the measurement and performance measurement process. Being very generic, the model can be used for various purposes. For instance, it can be used as a template to the design of specific measures and performance indicators. Moreover, it may act as a foundation to the development of a performance measurement system.
Our model translates the requirements that we have defined, which, in turn, derive from our findings in the measurement and performance measurement literature. As mentioned previously, some researchers pursue approaches that are not necessarily compatible with others. Thus, representing all the different proposals in a single model has not been possible. We have therefore been obliged to take decisions on one or the other subject, in order to be able to propose a coherent model.
Finally, we have presented a *Performance Measurement System Prototype Application* that builds upon our model proposal. We have described a generic, high-level architecture for performance measurement systems that relies on a three-tier architecture. The protoype application, which builds on the proposed architecture, is based on a simplified version of the measure and performance indicator model, and translates five basic use cases.

The prototype application that we have presented has allowed us to validate certain aspects of our model, while others have not been tested. The next logical step in our research constitutes in the testing of the model on a larger scale, and under more realistic circumstances. A project centred on process management or networking, for example, could provide the necessary context for measurements.
Quality of measurement is another possible research area, we have briefly invoked some aspects in section 2.5. For instance, from what 'angles' can quality of measurement be considered? And how can they be evaluated?
Golfarelli et al. [50] consider data streams as potential data sources for performance measurement systems. Exploring the implications of measurements on data streams, as well as their consequence on our model proposal, is another possible research direction.

Bibliography

[1] Lerina Aversano, Andrea De Lucia, Matteo Gaeta, Pierluigi Ritrovato, Silvio Stefanucci, and Maria Luisa Villani. Managing coordination and cooperation in distributed software processes: the GENESIS environment. *Software Process Improvement and Practice*, 9(4):239–263, 2004.

[2] Balanced Scorecard Collaborative. Balanced scorecard functional standards. Technical report, Balanced Scorecard Collaborative, `https://www.bscol.com/pdf/Standardsv10a.pdf`, May 2000.

[3] Balanced Scorecard Collaborative. BSC XML draft standards. Technical report, Balanced Scorecard Collaborative, `https://www.bscol.com/pdf/XMLStandards1.pdf`, March 2001.

[4] Daniele Ballarini, Marco Cadoli, Matteo Gaeta, Toni Mancini, Massimo Mecella, Pierluigi Ritrovato, and Giuseppe Santucci. *Cooperative Methods and Tools for Distributed Software Processes*, chapter Cooperative Software Development in GENESIS: Requirements, Conceptual Model and Architecture, pages 67–86. Franco Angeli, 2003.

[5] Victor R. Basili and Dieter Rombach. The Tame project: Towards improvement-oriented software envionments. *IEEE Transactions on Software Engineering*, 14(6):758–773, June 1988.

[6] Kent Bauer. The KPI profiler: CRM case study. *DM Review*, November 2004.

[7] Kent Bauer. KPIs – the metrics that drive performance management. *DM Review*, September 2004.

[8] Kent Bauer. KPI reduction the correlation way. *DM Review*, February 2005.

[9] Kent Bauer. KPIs: Avoiding the threshold McGuffins. *DM Review*, April, May, June, August 2005.

[10] Kent Bauer. KPIs: The seasonality conundrum. *DM Review*, September 2005.

[11] Jonathan D. Becher. Mitigating metrics madness: How to tell KPIs from mere metrics. *Cutter IT Journal*, 19(4):13–16, April 2006.

[12] Lamia Berrah. *L'indicateur de performance. Concepts et applications*. Cépadues-Editions, Toulouse, France, 2002.

[13] Umit S. Bititci. Modelling of performance measurement systems in manufacturing enterprises. *International Journal of Production Economics*, 42:137–147, 1995.

[14] Umit S. Bititci, Kepa Mendibil, Veronica Martinez, and Pavel Albores. Measuring and managing performance in extended enterprises. *International Journal of Operations & Production Management*, 25(4):333–353, 2005.

[15] Umit S. Bititci, Trevor Turner, and Carsten Begemann. Dynamics of performance measurement systems. *International Journal of Operations & Production Management*, 20(6):692–704, 2000.

[16] Michael Blaha and James Rumbaugh. *Modélisation et conception orientées objet avec UML 2*. Pearson, Paris, France, second edition, 2005.

[17] Xavier Blanc. *MDA en action. Ingénierie logicielle guidée par les modèles*. Eyrolles, Paris, France, 2005.

[18] Chantal Bonnefous. *Indicateurs de performance*, chapter La construction d'un système d'indicateurs pertinents et efficaces. Productique. Hermes Science, Paris, France, 2001.

[19] Mike Bourne, John Mills, Mark Wilcox, Andy Neely, and Ken Platts. Designing, implementing and updating performance measurement systems. *International Journal of Operations & Production Management*, 20(7):754–771, 2000.

[20] Mokrane Bouzeghoub and Veronika Peralta. A framework for analysis of data freshness. In *IQIS'04: Proceedings of the 2004 international workshop on Information quality in information systems*, pages 59–67, Paris, France, 2004.

[21] Bert R. Boyce, Charles T. Meadow, and Donald H. Kraft. *Measurement in Information Science*. Academic Press, San Diego, USA, 1994.

[22] Susan Brock, Danyal Hendricks, Stephen Linnell, and Derek Smith. A balanced approach to IT project management. In *SAICSIT'03: Proceedings of the 2003 annual research conference of the South African institute of computer scientists and information technologists on Enablement through technology*, pages 2–10, 2003.

[23] Marco Busi and Umit S. Bititci. Collaborative performance management: present gaps and future research. *International Journal of Productivity and Performance Management*, 55(1):7–25, 2006.

[24] Jean Bézivin. On the unification power of models. *Software and System Modeling (SoSym)*, 4(2):171–188, 2005.

[25] Vicente Aceituno Canal. ISM3 1.20, Information Security Management Maturity Model. `http://www.ism3.com/`, 2006.

[26] Cranfield School of Management, Centre for Business Performance. Catalogue of business performance measures. `http://www.som.cranfield.ac.uk/som/research/centres/cbp/products/catalogue.asp`.

[27] Karlene M. Crawford and James F. Cox. Designing performance measurement systems for just-in-time operations. *International Journal of Production Research*, 28(11):2025–2036, 1990.

[28] Howard Dresner. Business activity monitoring: BAM architecture. Gartner Symposium ITXPO, Cannes, France, November 2003.

[29] Wayne W. Eckerson. Development techniques for creating analytic applications. Report, The Data Warehousing Institute (TDWI), March 2005.

[30] Wayne W. Eckerson. Creating effective KPIs. *DM Review*, June 2006.

[31] Wayne W. Eckerson. Deploying dashboards and scorecards. Best practices report, The Data Warehousing Institute (TDWI), July 2006.

[32] Marc J. Epstein and Robert A. Westbrook. Linking actions to profits in strategic decision making. *MIT Sloan Management Review*, 42(3):39–49, 2001.

[33] Hans-Erik Eriksson and Magnus Penker. *Business Modeling with UML. Business Patterns at Work.* John Wiley & Sons, New York, USA, 2000.

[34] Stephen Fenstermaker, David George, Andrew B. Kahng, Stefanus Mantik, and Bart Thielges. METRICS: A system architecture for design process optimization. In *DAC'00: Proceedings of the 37th conference on Design automation*, pages 705–710, Los Angeles, USA, 2000.

[35] Norman Fenton. Software measurement: A necessary scientific basis. *IEEE Transactions on Software Engineering*, 20(3), March 1994.

[36] Pat Ferguson, Gloria Leman, Prasad Perini, Susan Renner, and Girish Seshagiri. Software process improvement works! Technical Report CMU/SEI-99-TR-027, Software Engineering Institute (SEI), November 1999.

[37] Alain Fernandez. *Les nouveaux tableaux de bord des managers*. Edition d'Organisation, Paris, France, 2005.

[38] Stephen Few. *Information Dashboard Design: the effective visual communication of data*. O'Reilly, Sebastopol, USA, 2006.

[39] Paul Folan and Jim Browne. Development of an extended enterprise performance measurement system. *Production Planning & Control*, 16(6):531–544, September 2005.

[40] Paul Folan and Jim Browne. A review of performance measurement: Towards performance management. *Computers in Industry*, 56(7):663–680, 2005.

[41] Cipriano Forza and Fabrizio Salvador. Assessing some distinctive dimensions of performance feedback information in high performing plants. *International Journal of Operations & Production Management*, 20(3):359–385, 2000.

[42] David S. Frankel. *Model Driven Architecture. Applying MDA to Enterprise Computing*. Wiley Publishing, Indianapolis, USA, 2003.

[43] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Professional Computing Series. Addison-Wesley, 2004.

[44] Félix García, Mario Piattini, Francisco Ruiz, Gerardo Canfora, and Corrado A. Visaggio. FMESP: Framework for the modeling and evaluation of software processes. *Journal of Systems Architecture: the EUROMICRO Journal*, 52(11):627–639, November 2006.

[45] Félix García, Manuel Serrano, Jose A. Cruz-Lemus, Francisco Ruiz, and Mario Piattini. Managing software process measurement: A metamodel-based approach. *Information Sciences*, 177(12):2570–2586, 2007.

[46] Gartner, Inc. The Gartner glossary of information technology acronyms and terms. Technical report, Gartner, Inc., `http://www.gartner.com/6_help/glossary/Gartner_IT_Glossary.pdf`, 2004.

[47] Alaa M. Ghalayini and James S. Noble. The changing basis of performance measurement. *International Journal of Operations & Production Management*, 16(8):63–80, 1996.

[48] Alaa M. Ghalayini, James S. Noble, and Thomas J. Crowe. An integrated dynamic performance measurement system for improving manufacturing competitiveness. *International Journal of Production Economics*, 48:207–225, 1997.

[49] Shlomo Globerson. Issues in developing a performance criteria system for an organisation. *International Journal of Production Research*, 23(4):639–646, 1985.

[50] Matteo Golfarelli, Stefano Rizzi, and Iuris Cella. Beyond data warehousing:

What's next in business intelligence? In *DOLAP'04: Proceedings of the 7th International Workshop on Data Warehousing and OLAP*, pages 1–6, Washington DC, USA, November 2004.

[51] Michael Goold and John J. Quinn. The paradox of strategic controls. *Strategic Management Journal*, 11(1):43–57, January 1990.

[52] Michel Grabisch, Sergei A. Orlovski, and Ronald R. Yager. *Fuzzy sets in decision analysis, operations research and statistics*, chapter Fuzzy aggregation of numerical preferences, pages 31–68. Kluwer Academic Publishers, Norwell, USA, 1998.

[53] Jean Harris. Performance models: Enhancing accountability in academe. *Public Productivity & Management Review*, 22(2):135–139, December 1998.

[54] Judith H. Hibbard, Jacquelyn J. Jewett, Mark W. Legnini, and Martin Tusler. Choosing a health plan: Do large employers use the data? *Health Affairs*, 16(6):172–180, 1997.

[55] Stefan Holmberg. A systems perspective on supply chain measurements. *International Journal of Physical Distribution & Logistics Management*, 30(10):847–868, 2000.

[56] Milagros Ibáñez. Balanced IT scorecard description version 1.0. Technical Report ESI-1998-TR-012, European Software Institute (ESI), May 1998.

[57] IEEE. IEEE standard glossary of software engineering terminology. Technical Report IEEE Std 610.12-1990, Institute of Electrical and Electronics Engineers (IEEE), 1990.

[58] Eric Jendrock, Jennifer Ball, Debbie Carson, Ian Evans, Scott Fordin, and Kim Haase. *The Java EE 5 Tutorial*. Addison-Wesley, third edition, 2006.

[59] Age Johnsen. What does 25 years of experience tell us about the state of performance measurement in public policy and management? *Public Money & Management*, pages 9–17, January 2005.

[60] Hubert Kadima. *MDA. Conception orientée objet guidée par les modèles*. Dunod, Paris, France, 2005.

[61] Stephen H. Kan. *Metrics and Models in Software Quality Engineering*. Addison-Wesley, second edition, 2003.

[62] Robert S. Kaplan and David P. Norton. The balanced scorecard - measures that drive performance. *Harvard Business Review*, 70(1):71–79, January-February 1992.

[63] Robert S. Kaplan and David P. Norton. *The Balanced Scorecard. Translating Strategy into Action.* Harvard Business School Press, Boston, USA, 1996.

[64] Robert S. Kaplan and David P. Norton. Using the balanced scorecard as a strategic management system. *Harvard Business Review*, 74(1):75–85, January-February 1996.

[65] Robert S. Kaplan and David P. Norton. Having trouble with your strategy? then map it. *Harvard Business Review*, 78(5):167–176, September-October 2000.

[66] Evangelia Kavakli and Pericles Loucopoulos. *Information Modeling Methods and Methodologies*, chapter Goal Modeling in Requirements Engineering: Analysis and Critique of Current Methods, pages 102–124. Advanced Topics in Database Research. IDEA Group Publishing, Hershey, USA, 2005.

[67] Vahé A. Kazandjian, Nikolas Matthes, and Karol G. Wicker. Are performance indicators generic? the international experience of the quality indicator project. *Journal of Evaluation in Clinical Practice*, 9(2):265–276, May 2003.

[68] Gerald Kotonya and Ian Sommerville. *Requirements Engineering: Processes and Techniques.* John Wiley & Sons, Chichester, England, 1998.

[69] Peter Kueng. Process performance measurement system: a tool to support process-based organizations. *Total Quality Management*, 11(1):67–85, January 2000.

[70] Peter Kueng, Thomas Wettstein, and Beate List. A holistic process performance analysis through a performance data warehouse. In *AMCIS'01: Proceedings of the 7th Americas Conference on Information Systems*, pages 349–356, Boston, USA, August 2001.

[71] Craig Larman. *UML 2 et les design patterns*. Pearson, Paris, France, third edition, 2005.

[72] Richard Lea and Brian Parker. The JIT spiral of continuous improvement. *Industrial Management & Data Systems*, 89(4):10–13, 1989.

[73] Maurizio Lenzerini. Data integration: a theoretical perspective. In *PODS'02: Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 233–246. ACM Press, 2002.

[74] John H. Lingle and William A. Schiemann. From balanced scorecard to strategic gauges: is measurement worth it? *Management Review*, pages 56–62, March 1996.

[75] Beate List and Karl Machaczek. Towards a corporate performance measurement

system. In *SAC'04: Proceedings of the 2004 ACM symposium on Applied computing*, pages 1344–1350, Nicosia, Cyprus, March 2004.

[76] Clemens Lohman, Leonard Fortuin, and Marc Wouters. Designing a performance measurement system: A case study. *European Journal of Operational Research*, 156(2):267–286, July 2004.

[77] Philippe Lorino. *Indicateurs de performance*, chapter La performance et ses indicateurs. Eléments de définition. Productique. Hermes Science, Paris, France, 2001.

[78] Leszek A. Maciaszek. *Requirements Analysis and System Design. Developing Information Systems with UML.* Addison-Wesley, 2001.

[79] Gus Manoochehri. Overcoming obstacles to developing effective performance measures. *Work Study*, 48(6):223–229, 1999.

[80] Bernard Marr. Automating intellectual capital – integrating measures for intangibles into corporate performance management. *International Journal of Learning and Intellectual Capital*, 1(3):304–316, 2004.

[81] Bernard Marr. Business performance management: current state of the art. Technical report, Cranfield School of Management, 2004.

[82] Jon Meliones. Saving money, saving lives. *Harvard Business Review*, 78(6):57–65, November/December 2000.

[83] Stephen J. Mellor and Marc J. Balcer. *Executable UML. A Foundation for Model-Driven Architecture*. Addison-Wesley, 2002.

[84] Kepa Mendibil and Jillian Macbryde. Designing effective team-based performance measurement systems: an integrated approach. *Production Planning & Control*, 16(2):208–225, March 2005.

[85] Pietro Micheli and Mike Kennerley. Performance measurement frameworks in public and non-profit sectors. *Production Planning & Control*, 16(2):125–134, March 2005.

[86] Tracy Miller and Sheila Leatherman. The national quality forum: A 'me-too' or a breakthrough in quality measurement and reporting? *Health Affairs*, 18(6):233–237, November/December 1999.

[87] Andy Neely. The performance measurement revolution: why now and what next? *International Journal of Operations & Production Management*, 19(2):205–228, 1999.

[88] Andy Neely and Mike Bourne. Why measurement initiatives fail. *Measuring Business Excellence*, 4(4):3–6, 2000.

[89] Andy Neely, Mike Gregory, and Ken Platts. Performance measurement system design: a literature review and research agenda. *International Journal of Operations & Production Management*, 15(4):80–116, 1995.

[90] Andy Neely, John Mills, Ken Platts, Huw Richards, Mike Gregory, Mike Bourne, and Mike Kennerley. Performance measurement system design: developing and testing a process-based approach. *International Journal of Operations & Production Management*, 20(10):1119–1145, 2000.

[91] Andy Neely, Huw Richards, John Mills, Ken Platts, and Mike Bourne. Designing performance measures: a structured approach. *International Journal of Operations & Production Management*, 17(11):1131–1152, 1997.

[92] Object Management Group (OMG). MDA guide, version 1.0.1. Technical Report omg/2003-06-01, Object Management Group (OMG), `http://www.omg.org/docs/omg/03-06-01.pdf`, June 2003.

[93] Oxford Reference Online. *The Oxford American Dictionary of Current English*. Oxford University Press, `http://www.oxfordreference.com`, 1999.

[94] Oxford Reference Online. *A Dictionary of Computing*. Oxford University Press, `http://www.oxfordreference.com`, 2004.

[95] Oxford Reference Online. *The Oxford Dictionary of English (revised edition)*. Oxford University Press, `http://www.oxfordreference.com`, 2005.

[96] Oxford Reference Online. *The Concise Oxford English Dictionary*. Oxford University Press, `http://www.oxfordreference.com`, 2006.

[97] Oxford Reference Online. *A Dictionary of Business and Management*. Oxford University Press, `http://www.oxfordreference.com`, 2006.

[98] Elaine Palmer and David Parker. Understanding performance measurement systems using physical science uncertainty principles. *International Journal of Operations & Production Management*, 21(7):981–999, 2001.

[99] Johann Pfanzagl. *Theory of Measurement*. Physica, Würzburg, Germany, 1968.

[100] Viara Popova and Jan Treur. A specification language for organisational performance indicators. In *IEA/AIE'2005: Proceedings of the 18th international conference on Innovations in Applied Artificial Intelligence*, pages 667–677, Bari, Italy, June 2005.

[101] Practical Software and Systems Measurement (PSM). Practical software and systems measurement. a foundation for objective project management, version 4.0. Technical report, Department of Defense and US Army, `http://www.psmsc.com/`, March 2003.

[102] Practical Software and Systems Measurement (PSM). Measurement specification template, version 2.0. Technical report, Department of Defense and US Army, `http://www.psmsc.com/SampleMeasures.asp`, year unknown.

[103] Sandeep Purao and Vijay Vaishnavi. Product metrics for object-oriented systems. *ACM Computing Surveys*, 35(2):191–221, June 2003.

[104] Andrea Rangone. An analytical hierarchy process framework for comparing the overall performance of manufacturing departments. *International Journal of Operations & Production Management*, 16(8):104–119, 1996.

[105] Fred S. Roberts. *Measurement Theory with Applications to Decisionmaking, Utility, and the Social Sciences*, volume 7 of *Encyclopedia of Mathematics and its Applications*. Addison-Wesley, Reading, USA, 1979.

[106] Paul Rouse and Martin Putterill. An integral framework for performance measurement. *Management Decision*, 41(8):791–805, 2003.

[107] Mehmet Sayal, Fabio Casati, Umeshwar Dayal, and Ming-Chien Shan. Business process cockpit. In *VLDB'02: Proceedings of the 28th International Conference on Very Large Data Bases*, pages 880–883, Hong Kong, China, August 2002.

[108] Douglas C. Schmidt. Model-driven engineering. *IEEE Computer*, 39(2):25–31, February 2006.

[109] Software Engineering Institute (SEI). Capability Maturity Model Integration (CMMI) for development, version 1.2. Technical Report CMU/SEI-2006-TR-008, Software Engineering Institute (SEI), `http://www.sei.cmu.edu/cmmi/`, August 2006.

[110] Il-Yeol Song, Kurt Yano, Juan Trujillo, and Sergio Luján-Mora. *Information Modeling Methods and Methodologies*, chapter A Taxonomic Class Modeling Methodology for Object-Oriented Analysis, pages 216–240. Advanced Topics in Database Research. IDEA Group Publishing, Hershey, USA, 2005.

[111] Patrick Suppes and Joseph L. Zinnes. *Handbook of Mathematical Psychology*, volume 1, chapter Basic Measurement Theory, pages 1–76. John Wiley & Sons, New York, USA, 1963.

[112] P. Suwignjo, U.S. Bititci, and A.S. Carrie. Quantitative models for performance

measurement system. *International Journal of Production Economics*, 64:231–241, 2000.

[113] E. Tapinos, R.G. Dyson, and M. Meadows. The impact of the performance measurement systems in setting the 'direction' in the university of warwick. *Production Planning & Control*, 16(2):189–198, March 2005.

[114] Axel van Lamsweerde. Goal-oriented requirements engineering: A guided tour. In *RE'01: Proceedings of the 5th IEEE International Symposium on Requirements Engineering*, pages 249–263, Toronto, Canada, August 2001.

[115] David Walters. Performance planning and control in virtual business structures. *Production Planning & Control*, 16(2):226–239, March 2005.

[116] Thomas Wettstein. *Gesamtheitliches Performance Measurement. Vorgehensmodell und informationstechnische Ausgestaltung*. Dissertation, Universität Freiburg, Schweiz, 2002.

Appendix A.

Measure Samples

We list here a number of measures that we have encountered during literature review. They are classified according to the domain they suit best. This collection is by no means complete, its sole purpose is to give an idea of what measures may be pertinent in a particular domain.
For a detailed discussion on measurement in software engineering, especially in software quality engineering, we refer the reader to Kan [61]. Measurement in information science, including information retrieval and information systems, is discussed by Boyce et al. [21]. Berrah [12] provides a detailed list of measures in the industry sector. Kaplan and Norton [63] discuss measurement in the fields of finance, customer, internal processes and learning and growth. The Centre for Business Performance of the Cranfield School of Management elaborated an extensive catalogue of business performance measures [26], covering fields such as stakeholder satisfaction, strategies, processes, capabilities, and stakeholder contributions. For a discussion on the ten most frequently used measures in the healthcare sector, we refer the reader to Kazandjian et al. [67].

Table A.1.: Measure samples

Domain	**Measure**
Customer relationship	Customer satisfaction (survey, focus groups)
	Customer retention
	Number of lost customers
	Customer acquisition
	Customer base growth per year
	Customer demographics
	Customer segmentation
	Customer profitability
	Active customer value
	Lost customer value
	Number of active/inactive customers
Customer service	Average waiting time in line
	Order process time
	Mean time response to a service call
	Average response time to complaints

continued on next page

Domain	Measure
	Number of complaints
Delivery	Delivery cost
	Delivery time
	Percentage of on-time deliveries
	Loss claims as percentage of revenue
	Damage claims as percentage of revenue
	Deliveries unmatched with invoice
	Delivery accuracy
	Number of late trucks (planning)
	Percentage of release errors
	Warranty and returns
	Time between order and delivery
Education & Training	Number of workshops held
	Number of people participating
	Percentage of people having followed training
	Number of hours per year/month of training per person
	Funds allocated for training programs
Environment	Air quality, air pollution index
	Noise level
	Quantity of waste
	Quantity of emissions
	Reuse/recycle rate of waste
	Energy consumption
	Water consumption
Finance	Revenues
	Expenses
	Profits
	Cash flow
	Return On Equity (ROE)
	Net Operating Assets (NOA)
	Weighted Average Cost of Capital (WACC)
	Net Operating Profit After Taxes (NOPAT)
	Return On Capital Employed (ROCE)
	Economic Value Added (EVA)
Healthcare	Total inpatient mortality
	Total perioperative mortality
	Primary Caesarean sections
	Repeat Caesarean sections
	Total Caesarean section frequency
	Vaginal births after Caesarean section
	Unscheduled re-admissions within 31 days for same or related condition
	Unscheduled returns to intensive care unit
	Unscheduled returns to the operating room
	Unscheduled returns to emergency department within 72h
Hospitality Industry	Occupancy
	Average daily rate

continued on next page

Domain	Measure
	Revenue per available room
Human Resources	Employee satisfaction
	Employee competence
	Employee turnover
	Count of open positions
	Overtime
	Absenteeism, lateness
Information retrieval	Precision
	Recall
	Noise
	Fallout
	Omission factor
	Specificity
Manufacturing	Number of units manufactured
	Manufacturing quality
	Percentage of products that fail test
	Number of defects
	Manufacturing cycle time
	Adherence to due date
	Machine utilization
	Machine reliability rates
	Total of maintenance time
	Operational cost per hour
	Process cost
	Manufacturing cost per unit
	Inventory cost
	Inventory turnover
	Raw material recycling rate
Marketing	Market share
	Campaign success
	Competitive ranking (through third-party organisations)
Products	Defect rate
	Product return rate (non-defective)
	Product return rate (defective)
Project Management	Actual number achieved on time
	Percentage achieved on time
	Schedule estimation accuracy
	Effort estimation accuracy
	Productivity (units of output per unit of effort)
	Number of changes in projects
	Planned number of critical dates
	Percent overtime
Research & Development	Product research cost
	Product development time
	Process time to maturity
	Time to market for new products
	Rate of new-product introduction

continued on next page

Domain	Measure
	Time to develop next generation
	Average time between subsequent innovations
	Number of new patents
Safety & Health	Number of accidents reported
	Number of man-days lost
Sales	Actual bookings $
	Average daily bookings $
	Average daily bookings (units)
	Sales planned $
	Sales planned (units)
	Actual bookings % of sales planned
	Sales forecast $
	Sales forecast (units)
	Actual bookings % of sales forecast
	Forecast accuracy
	Percent of sales from new products
	Sales growth
	Average selling price
	Number of orders
	Order amounts
Software Development Processes	Design review coverage
	Code inspection coverage
	Inspection effort
	Average defects per inspection
	Defect arrivals by time interval
	Testing coverage
	Number of defects per executed test case
	Defect density during testing
	CPU utilisation during test
	System crashes and hangs during test
	Defect backlog, number of unsolved defects
	Defect removal effectiveness
	Defect containment effectiveness
	Percent of function integration to system library
	Fix backlog, number of open problems
	Mean age of open problems
	Mean age of closed problems
	Backlog management index
	Fix response time
	Percent delinquent fixes
	Fix quality
	Cost of fixing problems
Software Products	Size (lines of code (LOC), function points, classes)
	Number of classes
	Average class size LOC
	Average number of methods per class
	Average method size LOC

continued on next page

Domain	Measure
	Weighted methods per Class (WMC)
	Average number of instance variables per class
	Depth of inheritance tree (DIT)
	Average number of comment lines (per method)
	Number of times a class is reused
	Response for a class (RFC)
	Lack of cohesion of methods (LCOM)
	Coupling between object classes (CBO)
	Number of children of a class (NOC)
	Lack of cohesion on methods (LCOM)
	Defect density
	Mean time to failure (MTTF)
	Mean time to repair (MTTR)
	Failure rate
	Percent of system availability
Technical support	Number of support calls
	Number of resolved cases
	Call durations
	Customer satisfaction
Telecommunication Networks	Call setup delay
	Call setup success rate
	Handover failures
	Round-trip time
	One-way delay
	Packet loss ratio
	Bandwidth utilization
	Network downtimes (scheduled or unscheduled)
Web Sites	Number of visitors
	Number of new visitors
	Average number of pages visited
	Average visit duration

Appendix B.

Performance Measurement Frameworks

Table B.1.: Performance measurement frameworks

Author / Source	**Type**	**Framework**
DuPont (1920s)	Structural	DuPont pyramid
Basili & Rombach [5] (1988)	Procedural	Goal/Question/Metric method
Sink & Tuttle (1989)	Procedural	Six-step process model
Keegan et al. (1989)	Structural	Performance Measurement Matrix
Dixon et al. (1990)	Procedural	Performance Measurement Questionnaire
Fitzgerald et al. (1991)	Structural	Results and Determinants Framework
Lockamy (1991)	Structural	Four theoretical performance measurement system models
Lynch & Cross (1991)	Structural	Performance Pyramid
Lynch & Cross (1991)	Procedural	Ten-step process model
Azzone et al. (1991)	Structural	Internal/external configuration time framework
Kaydos (1991)	Procedural	Kaydos' framework
Wisner & Fawcett (1991)	Procedural	Nine-step process model
Kaplan & Norton [62] (1992)	Structural	Balanced Scorecard
Kaplan & Norton (1993)	Procedural	Eight-step process model
Bradley (1996)	System	AMBITE performance measurement cube
Brown & Score (1996)	Structural	Brown's framework
Kaplan & Norton [63, 64] (1996)	System	Balanced Scorecard
Ghalayini et al. [48] (1997)	System	Integrated Dynamic Performance Measurement System
European Foundation for Quality Management (EFQM) (1999)	Structural	EFQM Model
Neely et al. [90] (2000)	Procedural	Ten-step process model
Medori & Steeple (2000)	System	Medori and Steeple's model
Hudson et al. (2001)	Procedural	SME performance measurement framework
University Bordeaux (2001 ?)	System	ECOGRAI method

continued on next page

Author / Source	Type	Framework
Wettstein [116] (2002)	Procedural	Wettstein's model
Neely et al. (2002)	Structural	Performance Prism
Yeniyurt (2003)	Structural	Framework for multi-national companies
PSM [101] (2003)	System	Practical Software & Systems Measurement methodology
Rouse & Putterill [106] (2003)	Structural	Integrated performance measurement framework
Mendibil & Macbryde [84] (2005)	Procedural	Nine-step process model for team-based performance measurement

Appendix C.

Performance Measurement Systems

C.1. Academic Systems

Table C.1.: Academic performance measurement systems

Author / Source	System Name	System Description
Lohman et al. [76]		Case study on the design of a Performance Measurement System in a major company producing and selling sportswear. The study resulted in a Excel-based prototype system. Aspects addressed in the study: impact of existing measures on the development of a PMS, aggregation of PIs, standardised expression of metrics (metrics dictionary).
Fenstermaker et al. [34]	METRICS	METRICS is a system that measures the semiconductor design process in order to be able to optimise it. The system gathers artefact, process and communication data from the design tools (via wrappers or embedded scripts) and stores them in a metrics data warehouse. Various reports can be accessed from web browsers.
Sayal et al. [107]	Business Process Cockpit	The Business Process Cockpit is a tool that supports monitoring, analysis, and management of business processes. An Extract, Transfer, and Load (ETL) application collects data from the log and loads them into a Process Data Warehouse. The user can visualise process execution data and metrics according to different focus points and perspectives. Moreover, alerts can be defined, as well as actions to be executed as soon as an alert is detected.

continued on next page

Author / Source	System Name	System Description
Ballarini et al. [4], Aversano et al. [1]	GENESIS	The GENESIS platform is the outcome of a research project aiming at designing and developing a non-invasive system to support software engineering processes in a highly distributed environment. It provides a module, called GEM (GENESIS Measurements), for monitoring the project enactment and collecting process metrics. GEM is in charge of periodically (batch mode) collecting metrics during the process and of presenting synthetic reports about the project status (on demand mode), furthermore, it allows the generation of alarm events.
Kueng et al. [70]	Performance Cockpit	The system relies on a multi-dimensional data warehouse where each cube has its own measure dimension. A second database stores auxiliary data including access rights and target values. The web-based dashboard shows actual and target values, as well as the performance gap which is represented by 'traffic lights'. Details of the data is available for each indicator for further analysis.
García et al. [44, 45]	GenMETRIC	GenMETRIC is a tool that allows the definition, calculation and visualisation of software metrics. Measures are defined in measurement models and are calculated on domain models, the models of the software entities under consideration (both models derive from metamodels). The metamodels and models are stored in a repository as XMI documents.

C.2. Commercial and Open-Source Systems

Table C.2.: Commercial and open-source performance measurement systems

Editor	Product	Description
Advizor Solutions `http://www.advizorsolutions.com`	Visual Discovery	Analysis and dashboards
arcplan `http://www.arcplan.com`	arcplan Enterprise	Analysis and dashboards
BusinessObjects `http://www.businessobjects.com`	BusinessObjects XI	Business intelligence suite
BusinessObjects `http://www.xcelsius.com`	Crystal Xcelsius	MS Excel-based dashboards

continued on next page

Editor	Product	Description
Cognos http://www.cognos.com	Cognos	Business intelligence suite
Corda Technologies http://www.corda.com	CenterView	Dashboards
eclipse http://www.eclipse.org/dash	Project Dash	Project monitoring
Epitome Systems http://www.epitomesystems.com	Enterprise Productivity Management Software Suite	Business intelligence suite
Hyperion / Oracle http://www.hyperion.com	Hyperion System	Business intelligence suite
IBM http://www-306.ibm.com/software/genservers/portal	WebSphere Dashboard Framework	Dashboards
iDashboards http://www.idashboards.com	iDashboards	Dashboards
Informatica http://www.informatica.com	PowerCenter	Data integration and dashboards
iQ4bis http://www.q4bis.com	iQ4bis	Analysis and dashboards
JasperSoft http://www.jaspersoft.com	Business Intelligence Suite	Open source business intelligence suite
LogiXML http://www.logixml.com	Logi Platform	Reporting and dashboards
MicroStrategy http://www.microstrategy.com	MicroStrategy	Business intelligence suite
Noetix http://www.noetix.com	Noetix Platform	Reporting and dashboards
Pentaho http://www.pentaho.com	Pentaho Open BI Suite	Open source business intelligence suite
Principa http://principa.net	Business DashBoard	Dashboards
QPR http://www.qpr.com	QPR ScoreCard	Dashboards
SAS http://www.sas.com	SAS Business Intelligence	Business intelligence suite
SourceForge https://sourceforge.net	Tracker	Project monitoring
SpagoBI http://spagobi.eng.it	SpagoBI	Open source business intelligence suite
SPSS http://www.spss.com	ShowCase Suite	Analysis, reporting and dashboards
Theoris http://www.theoris.com	Theoris Vision Software	Dashboards
Visual Mining http://www.visualmining.com	NetCharts	Dashboards

Appendix D.

Prototype Artefacts

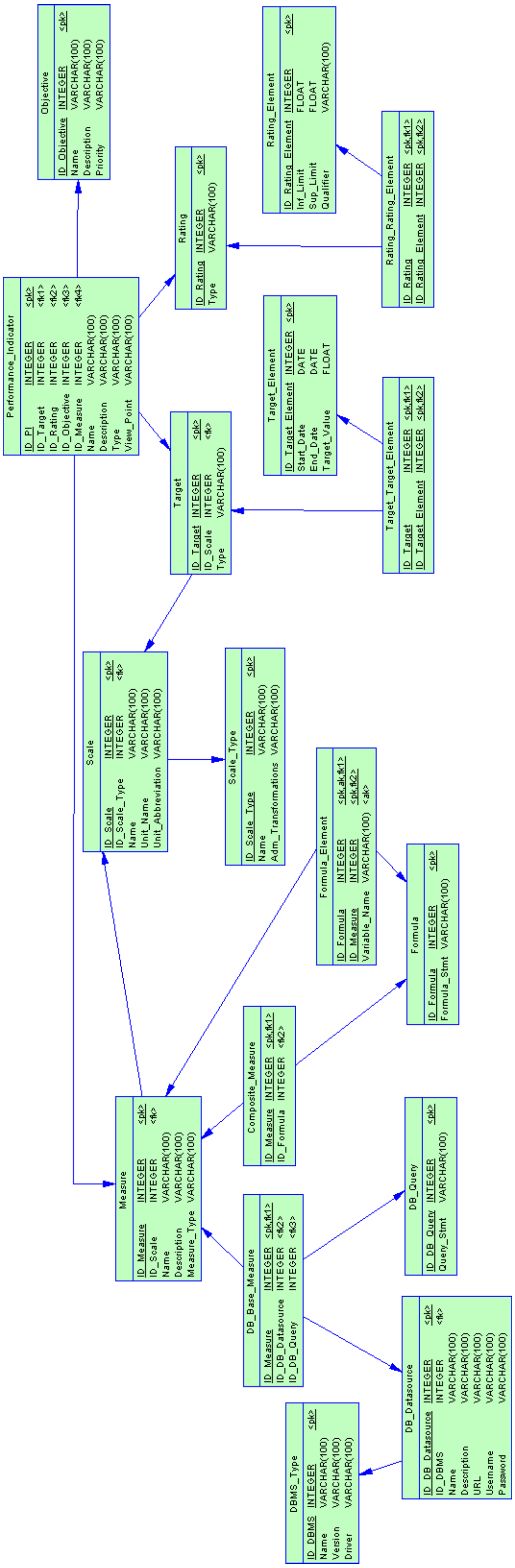


Figure D.1.: Database schema of the metadata repository