MetricM: A modeling method in support of the reflective design and use of performance measurement systems*

Stefan Strecker† Ulrich Frank‡ David Heise§ Heiko Kattenstroth¶

Information Systems and Enterprise Modelling Research Group
Institute for Computer Science and Business Information Systems,
University of Duisburg-Essen, Universitaetsstr. 9, 45141 Essen, Germany

Abstract

Performance indicators play a key role in management practice. The existence of a coherent and consistent set of performance indicators is widely regarded as a prerequisite to making informed decisions in line with set objectives of the firm. Designing such a system of performance indicators requires a profound understanding of the relations between financial and non-financial metrics, organizational goals, aspired decision scenarios, and the relevant organizational context—including subtleties resulting from implicit assumptions and hidden agendas potentially leading to dysfunctional consequences connected with the ill-informed use of performance indicators. In this paper, we investigate whether a domain-specific modeling method can address requirements essential to the reflective design of performance measurement systems, and which structural and procedural features such a method entails. The research follows a design research process in which we describe a research artifact, and evaluate it to assess whether it meets intended goals and domain requirements. In the paper, we specify design goals, requirements and assumptions underlying the method construction, discuss the structural specification of the method and its design rationale, and provide an initial method evaluation. The results indicate that the modeling method satisfies the requirements of the performance measurement domain, and that such a method contributes to the reflective definition and interpretation of performance measurement systems.

Key words: Performance Measurement ; Enterprise Modeling ; Metamodeling ; Domain-Specific Modeling Language ; Method Engineering ; Design Research

*Preprint submitted to Information Systems and e-Business Management (Springer).

†stefan.strecker@uni-due.de
‡ulrich.frank@uni-due.de
§david.heise@uni-due.de
¶heiko.kattenstroth@uni-due.de

1 Introduction

Performance indicators are deeply entrenched in the long and intricate history of the measurement of organizational performance. Popularized in the aftermath of the "Management By Objectives" interpretation of Drucker (1954)'s "The Practice of Management" and picked up by the performance measurement "movement" (Eccles, 1991), performance measures appear to be *the* key tool in the practicing manager's toolkit:[1] The availability of a coherent and consistent set of performance indicators is regarded as a prerequisite to making informed decisions in line with organizational goals (Fortuin, 1988; Epstein and Manzoni, 1998). Linking performance indicators to incentive systems further attempts to establish management control (Simons, 1995; Simons et al, 2000). Originally focused on the financial results of the firm, it has repeatedly been proposed that firms should complement financial indicators (e.g. turnover, return on capital employed) with non-financial indicators (as related to time, quality and flexibility) to form a comprehensive performance measurement system as associated with the *SMART Pyramid* (Lynch and Cross, 1991), the *Balanced Scorecard* (Kaplan and Norton, 1992, 1996a) and *Strategy Maps* (Kaplan and Norton, 2004).

Performance indicators are constructs designed to create a model of organizational performance appropriate for a specific purpose. They are conceived by purposeful abstraction based on the plausible assumption that managing large organizations requires the reduction of complexity in order to avoid information overload (Lebas and Euske, 2007). In addition, managerial use of indicators is motivated by two further assumptions. First, setting and pursuing objectives is often regarded as an essential prerequisite of rational action (Simon, 1964). Second, it is a widespread belief that outcomes of a course of action affecting the achievement of an organizational goal need to be measurable to permit decision making and management control (Ridgway, 1956). Management practice has—as it appears, inevitably—adopted the use of performance indicators. Performance measurement systems seem to pervade most medium-sized and large organizations (e.g. Speckbacher et al, 2003).

[1] In line with Lebas and Euske (2007, 128), *"[w]e prefer the word 'indicator' to the more traditional one of 'measure'. A measure often implies precision; it is usually well-defined, and in similar circumstances its numerical value should be the same. An indicator may be less precise, but meaningful; indicators tend to allow for more timely and sensitive signals".* The term "measure" rather than "indicator" is common in performance measurement literature (Kaplan and Norton, 1992; Neely et al, 1997; Bourne et al, 2005). The latter underlines that a numerical representation is deliberately chosen—according to some rule—to indicate the state of some entity that is being evaluated (Nagel, 1931; Smith Stevens, 1959). Managers often refer to "Key Performance Indicators" (KPI) suggesting a deliberate selection from among a wide range of conceivable performance indicators for a specific scope (Hope, 2007). Also note the different connotations of the term "measure" when used as a verb or noun. "Measure" as a noun has a meaning of "an official action that is done in order to achieve a particular aim" (Oxford Advanced Learner's Dictionary, 7th edition) which complicates interpretation in the context of our study. Hence, for the sake of terminological clarity, we prefer to strictly distinguish between actions leading to some outcome and indicators designed to *measure* some feature of an outcome. Thus, performance indicator system and performance measurement system will be used interchangeably.

Though performance indicators may be an effective instrument for decision making and for management control, they can bring with them serious drawbacks (Perrin, 1998). First, indicators may not adequately measure contributions to the achievement of goals they are supposed to operationalize (Nørreklit et al, 2006, 2007). As a consequence, they may compromise decision making. Second, intentions connected with organizational indicator use may be in conflict. This can lead to misunderstandings and to staff acting in ways contrary to organizational goals. Hence, indicators may promote actions incoherent or even contradictory with respect to organizational goals. In addition to encouraging *intended* actions, incentives linked to indicators may also have dysfunctional consequences (Ridgway, 1956). Lastly, there are doubts that all aspects relevant for organizational decision making can be represented by performance indicators without an unacceptable degree of bias (Moers, 2005). In summary, performance indicators are ambivalent concepts. On the one hand, they may be beneficial aids that help to cope successfully with complexity in organizations and that contribute to reasoned justification. On the other hand, they may be a source of misconceptions and bias that is a threat to effective, goal-oriented organizational action (Townley et al, 2003 provide an illustrative example in this regard). It is this particular ambivalence that calls for the thoughtful, reflected and rational—in short, the reflective—design and subsequent use of performance indicators and performance measurement systems (for an ethical reasoning, see Rosanas and Velilla, 2005; Rosanas, 2008). The design of performance indicators and performance measurement systems is extensively discussed in literature (for an overview, see e.g. Bourne et al, 2003; Chenhall and Langfield-Smith, 2007). However, *methodical support* for the *reflective* design and use of performance measurement systems has not yet received particular attention in literature (Nørreklit et al, 2007).

Organizational performance is "a relative concept, requiring judgment and interpretation" (Lebas and Euske, 2007, 136) on the part of both the stakeholders involved and prospective users. Interpretation and judgment presupposes a shared understanding of organizational performance, which in turn implies a common conceptual framework of constituent terms and their semantics. One of the constituents of such a framework, the organizational context, is of particular importance to accurate interpretation (Neely et al, 1997). Thus, performance indicator design has, in particular, to account for the organizational goal the indicator is designed to represent, the resources and processes the indicator relates to, and the organizational roles involved in its use (Neely et al, 1995; Bourne et al, 2000; Tuomela, 2005). Moreover, relations among (non-financial and financial) indicators must be specifically designed to develop a performance measurement system (Eccles and Pyburn, 1992).

Present approaches to the design of performance measurement systems include only limited consideration of organizational context (e.g. by focusing on business processes only) and largely ignore the visual language (i.e. the actual graphical representation and corresponding symbolism) of performance measurement systems (for an exception, see Abernethy et al, 2005). At the same time, recent experimental evidence suggests "that firms should carefully consider how to [graphically] present and organize measures to get the intended effect on performance evaluations" (Cardinaels and van Veen-Dirks, 2010, 565). In particular, the semantics of concepts used in diagrammatic representations of performance measurement systems are rarely, if ever, elaborated upon. Rather, it appears to be presumed that prospective users will need the flexibility to adapt the diagrammatic representations to their own liking and to supplement them with textual explanations if needed. The resulting free-form drawings come, however, at a price of increased difficulty when interpreting the performance measurement system, of unsafe ambiguity of relations between indicators, and, lastly, of the danger of divergent interpretation which may lead to irritations and even failing to achieve goals, for instance, when a course of action is chosen and carried out that neglects the relevant organizational context.

These observations motivate research on conceptual modeling methods for performance measurement (e.g. Pourshahid et al, 2007; Frank et al, 2008; Popova and Sharpanskykh, 2010). Modeling methods in general and those based on a graphical domain-specific modeling language (DSML) in particular promise to support creating and interpreting performance measurement systems effectively and efficiently by providing differentiated semantics of dedicated modeling concepts and corresponding descriptive graphical symbols that further comprehensible performance measurement systems, and by making the conceptual model of a performance measurement system accessible to partly automated, tool-supported analyses that assist in its design, maintenance, and evaluation. It also provides a conceptual foundation for the design of corresponding information systems—for instance, management dashboards (Palpanas et al, 2007).

The present work follows a design research process to develop a conceptual modeling method for organizational performance measurement, and to investigate how its structural and procedural features can satisfy essential requirements in the performance measurement domain. The method, MetricM, consists of a domain-specific modeling language, MetricML, and a corresponding process model to prescribe the use of language concepts for performance measurement applications. MetricM is integrated with an enterprise modeling approach to benefit from the reuse of modeling concepts representing essential organizational context. The method's main purpose is to guide, promote and cultivate the reflective design, use and interpretation of performance indicators and performance measurement systems. Thus, MetricM is aimed at stimulating and facilitating communication among stakeholders involved in the design and use of performance measurement systems. It also aims to increase the transparency of indicator matters, specifically by linking models of indicators with models of the corresponding action system, hence improving traceability of interrelations among indicators and the relevant organizational context. This linkage is intended to contribute to the substantiation of performance indicators and to the identification of (important yet often overlooked) dependencies. Ultimately, MetricM is aimed at facilitating a reflected interpretation of a performance measurement system within its organizational setting. Earlier work on MetricM, in particular a predecessor to MetricML, is discussed in Frank et al (2008, 2009).

The next section reviews related work on conceptual modeling of performance indicators and performance measurement systems. Section 3 discusses the epistemological conception underpinning the research on MetricM. Section 4 outlines the theoretical background informing the method design and establishes domain-specific requirements as well as key domain-specific terminology. The design rationale of the structural specification of the method, its metamodel, is discussed in Section 5 along with procedural guidelines on the application of language concepts. An initial method evaluation is provided in Section 6. The paper concludes with a discussion of findings in Section 7.

2 Related work

Among the first to consider performance indicators as part of a conceptual modeling approach was Österle (1995, 112ff.). The metamodel of his PROMET method includes a "Performance Indicator" metatype with two named meta-associations. It "belongs" to a metatype "Process" and "operationalizes" a metatype "Critical success factor" (Bach et al, 1996, 270). In addition, the PROMET literature illustrates the use of these modeling concepts at type level by listing a number of exemplary indicator types such as (average) cycle time and rate of customer complaints as well as through an elaborate process model guiding the use of these concepts (Österle et al, 1996). Aichele (1997) extends the ARIS method (Scheer, 1992, 2000) by a differentiated indicator typology and a comprehensive list of potential indicators for business functions. Kronz (2005) refines his work with respect to business process indicators as part of the Event-Driven Process Chain (EPC) notation. Though, neither Aichele (1997) nor Kronz (2005) presents a language specification and—as far as it can be concluded from the language application they provide by example—only propose one type of relation between indicators (i.e. subordination).

Several early stage proposals of domain-specific languages for performance indicator modeling have been published. Referring to prior work by Karagiannis et al., Ronaghi (2005) introduces a metamodel of more than 20 metatypes ranging from "Incentive System" to "Target" including metatypes for representing indicator types (e.g. "P-Indicator" and "Elementary P-Indicator"). Since attributes are missing from the metamodel, the semantics of concepts are left to interpretation by the language user. Further performance measurement extensions to the EPC approach have recently been proposed by Korherr and List (2007a,b). Their work extends an EPC metamodel with a "Measure" metatype from which three metatypes "Quality", "Cost", and "Cycle Time" are specialized. Accordingly, three notation elements are introduced to represent the latter. For each of these three specialized metatypes, two meta-attributes specify further semantics (e.g. "Quality" is described by "maxComplaints: int" and "avgComplaints: int") (Korherr and List, 2007a, 289). The only named meta-association "quantifies" links the metatype "Measure" with the metatype "Process Goal" (Korherr and List, 2007a, 292). Neither Ronaghi (2005) nor Korherr and List (2007a,b) account for further organizational context or an indicator rationale.

Both only briefly mention the semantics of language concepts and a corresponding graphical notation.

A more elaborate approach to modeling performance indicators is presented by Popova and Sharpanskykh (2010). They introduce a modeling method aimed at formalization of performance indicators and their relations to support simulation, verification and validation (pp. 505 and 507). The corresponding "meta model for the *performance-oriented view*" (Popova and Sharpanskykh, 2010, 511) is illustrated in an unspecified notation showing concepts and their relations; semantics of modeling concepts are specified by natural language documentation and, partly, by an order-sorted predicate logic. The corresponding process model comprises guidelines to elicit performance indicators and their relations as well as an algorithm to compute "the satisfaction (degree of satisficing) of a goal" (Popova and Sharpanskykh, 2010, 514). Furthermore, the corresponding LEADSTO modeling tool is described. The modeling concepts include the meta types "PI" (Performance Indicator) and "Goal". Organizational context is represented by business processes, an agent and a role concept. Different from the present work, the authors do not discuss a domain-specific visual language—a notation and corresponding diagram types—for representing indicator systems. Their case study example illustrates the application of their modeling language using circles as symbols for indicators and labeled directed edges as symbols for relations. Performance indicators types are further visualized using a textual template.

Pourshahid et al (2007, 2008) extend the User Requirements Notation (URN)—an ITU-T Recommendation incorporating two notations, the Goal-oriented Requirements Language (GRL) and Use Case Maps (UCM)—with concepts for indicator modeling. Their framework specifies a meta model in which the meta type "Indicator" is further refined by meta attributes including, e.g., "isTimeMeasure", "isCostMeasure", and "isQualityMeasure" (i.e. type differentiation is based on meta attributes) and can be categorized according to an "IndicatorGroup" (Pourshahid et al, 2007, 84). "Indicator" is associated with "Performance Goal" and "Process" (Pourshahid et al, 2008, 7). The authors do not provide further concepts representing organizational context. A hexagon is introduced as the only graphical notation—representing an indicator presumably at type level along with suggestions for several diagram types including a "Performance model" (Pourshahid et al, 2008, 11–12).

The reviewed prior work illustrates the diversity of issues related to indicator modeling. However, earlier work does not consider the reflective design and use of performance indicators and performance measurement systems. Table 3 in Section 6 summarizes key concepts in related work and identifies areas of improvement over existing approaches in the light of domain-specific requirements. Research on MetricM builds upon prior approaches in that earlier conceptualizations, in particular by Pourshahid et al (2007, 2008) and Popova and Sharpanskykh (2010), are reconceptualized and extended by further modeling concepts and a corresponding graphical notation.

3 Research method

The artifact designed in this research is a modeling method; a linguistic artifact consisting of a conceptual modeling language and a process model to guide the use of language concepts (Wand et al, 1995; Frank, 2002; Wand and Weber, 2002). The main challenge for conceptualizing research aimed at the development of modeling methods as artifacts is their justification according to scientific standards (Frank, 1998, 2006a,b). The success of using a modeling method depends on various factors—qualification, previous experience with other languages, having time to learn the method, and attitude toward new methods—which not only vary between different groups but also within a group in time. Furthermore, we assume that prospective method users at present do not have a clear understanding of current and future applications of conceptual model-based performance measurement methods and are, hence, not *yet* able to evaluate their practical utility. This does not preclude empirical studies on the practical utility of the MetricM method in the future; at a point in time when method use has created a knowledgeable and large enough user base. Presently, however, field studies to test the newly conceived modeling method MetricM are not satisfactory due to subject contingency (Frank, 2005, 153).

The present work on MetricM is therefore grounded on a research method configured for the epistemological particularity of research on modeling methods (Frank, 2006b). The particular configuration of the research method suggests two main guidelines for the research process: multi-criteria justification and transparency of assumptions. Multi-criteria justification is based on the belief that there are various approaches available to substantiate an assumption. The selection depends on the theory of truth that is regarded as suitable, and the feasibility of corresponding justification procedures. Justification procedures include empirical tests (correspondence theory), discursive evaluation (consensus theory) and coherence with an existing body of accepted knowledge (coherence theory). The configuration approach provides criteria to guide the selection of justification procedures (Frank, 2006b, 48ff.). Combining the selected justification procedures results in the configuration of a research method that accounts for the epistemological particularity of the corresponding research. Note that the most appropriate justification procedure may not be practicable, perhaps because of the lack of time or resources or some other obstacle. In this case, the configuration approach recommends applying the second or third best option. Applying such a configuration approach does not guarantee a convincing justification. It does, however, contribute to an incremental justification and supports the further evaluation of the artifact by making it clear where its justification is still not satisfactory.

Transparency means that all non-trivial assumptions about the method design are identified throughout the research process. This pertains to requirements, design decisions, and the evaluation of the artifact against the requirements. To guide the method's development, its purpose and design goals need to be substantiated by requirements. If a requirement is not obvious or deducted from established knowledge, it is based on an assumption. The construction of the method or parts of it, in this case of a domain-specific modeling language, implies choices of design alternatives. Again, the assumptions underlying non-trivial design decisions are to be made explicit. Finally, the resulting method is evaluated by comparing its features against the design goals and requirements. In some cases, checking if a requirement is met will be straightforward. Meeting a requirement may be as simple as the presence of a certain feature of the artifact, for example. In other cases, however, evaluation requires assumptions; as is particularly the case with respect to requirements that relate to user acceptance or perceived benefit (Frank, 2006b, 55).

The justification procedures used in the present research are a combination of discursive evaluation and the application of the coherence theory of truth, i.e. substantiating assumptions by reference to a body of literature. Empirical tests are not included due to subject contingency and lack of feasibility at present. Note that this does not mean that empirical studies are unsuitable for testing modeling methods in general. If modeling methods are more widely used in the performance measurement domain, it can be more promising to pursue an empirical evaluation. Discursive justification in its ideal form would involve a rational discourse within a group of outstanding experts. A consensus on the truth value of a proposition would then be regarded as a satisfactory—albeit preliminary—test. This study applies a relaxed form of discursive evaluation. It starts by establishing high-level assumptions on design goals, which are likely to be agreed upon by many with knowledge of the domain of performance measurement. It proceeds to analytically deduce more specific requirements, which can be assumed to elicit consensus, and which are substantiated by the existing body of literature. In some cases, this approach will produce only weak justifications—a result which may be explained by the idiosyncrasy of the topic. In order not to impair the paper's readability, not every assumption will be explicitly marked as such.

Research on MetricM adapts a method engineering approach (Rossi et al, 2004; Rolland, 2007; Henderson-Sellers and Ralyté, 2010) to the construction of a modeling method as artifact (Frank, 2010a). In particular, the method construction is aimed at supporting a range of application projects—not a single, particular project as is typically the case in domain-specific (modeling) language literature (e.g. Kelly and Tolvanen, 2008; Strembeck and Zdun, 2009). Furthermore, the MetricM method design targets integration with an enterprise modeling method to benefit from reuse of existing modeling concepts and procedural guidelines. Developing a domain-specific modeling language in this context presupposes reconstructing key technical terms of the targeted domain and their semantics (Ortner, 2008). One widespread approach to conceptual reconstruction—and the one we follow here—is to review, analyze and interpret pertinent literature in the field under consideration (e.g. Neely, 2007; Kaplan, 2010; Taticchi, 2010). Reconstruction of technical terminology is an iterative process involving more than the identification of candidate (meta) concepts, their attributes and relations. Instead it requires, for instance, the identification and resolution of terminological ambiguity and truncation, which may imply the introduction of additional abstractions. That in turn may require the shaping of their semantics. This implies the (re-)interpretation of observed terms and concepts and leads to design abstractions appropriate for specific analyses and applications. The underlying method engineering approach is therefore driven by devising and analyzing domain-specific application scenarios describing, among others, prototypical model-based analyses (Frank, 2010a).

In the light of idealized design research processes (e.g. Verschuren and Hartog, 2005; Peffers et al, 2007; Österle et al, 2010), the present work reports on the phases of assumptions and requirements (phase 2 in Verschuren and Hartog (2005)'s model; corresponds with Section 4), structural specification (phase 3; corresponds with Section 5), prototype and evaluation (phases 4 and 6 corresponds with Section 6).

4 Domain analysis

4.1 Theoretical background

The claim for the reflective design and use of performance measurement systems links to fundamental issues in organizational theory, economic sociology, organizational psychology, and in the philosophy of science, for instance: What constitutes the quality of an indicator with respect to its support for decision making and management control? How can the quality of an indicator be assessed—and promoted? How does a performance indicator or a performance measurement system affect decisions and actions, i.e. stakeholder behavior, in an organization? Which factors limit the purposeful use of performance indicators? The following discussion briefly highlights the discourse on organizational performance indicator use.

As opposed to measurements derived from the physical world, objects of measurement in the realm of organizational performance are (latent) constructs (e.g. customer satisfaction, employee loyalty or return on capital employed). Similar to a measuring instrument, a performance indicator is directed at measuring a certain aspect of organizational performance with respect to a certain reference object (e.g. an organizational unit, a project, a product). Thus, a performance indicator is based on the hypothesis that it appropriately represents that particular aspect of organizational performance (Pike and Roos, 2007). Hence, an indicator involves an epistemological claim: Only if its validity is satisfactory, can it serve its purpose. An indicator is valid if it actually measures the manifestations of the targeted aspect; in other words, if it depends exclusively on that aspect. Validity implies objectivity and reliability (Edwards and Bagozzi, 2000). An indicator is objective if its measurement does not depend on the judgment of a particular person. It is reliable if repeated measurements will produce the same result. Although the question of how to judge the validity of a measurement is well-known in the philosophy of science, it has, nevertheless, no straightforward answer. Performance indicator validity depends on the truth of the hypothesis underlying the indicator or, in other words, on how well it can be justified.

Three essential theories of truth are of practical relevance to justify a hypothesis underlying a performance indicator (see Section 3; for a more elaborate discussion, see Frank, 2006b). The *correspondence theory of truth* is based on empirical evidence. For example, if past experience has shown that there is a strong correlation between the level of training a sales representative has received and the satisfaction of the customers he or she served, this would contribute to the justification of the level of training as a possible indicator of customer satisfaction. According to the *coherence theory of truth*, a hypothesis should not contradict an affirmed body of knowledge. With respect to the design of indicators, it recommends, among other things, analyzing whether the conception of an indicator is in line with existing indicators. Finally, the *consensus theory of truth* recommends rational discourses to judge the truth of a statement or, rather, how appropriate it is. It suggests identifying all underlying assumptions and criticizing them regardless of who made them in an open and free atmosphere characterized by a joint desire to achieve the best result. The more precise the conception of a reference object the better is the chance to assess the validity of an indicator. Often, however, only a vague conception exists, for instance, performance of an organizational unit, performance of a manager, customer satisfaction etc. In these cases, it is very likely that the borderline between an indicator and its reference object gets blurred. In the end, the performance of an organizational unit is not measured by the indicator "service quality"; rather "service quality" is regarded as the unit's performance. This leads to two further aspects of performance indicators: the limits of formalization and the social construction of reality. Conceptualizing an indicator in a way that allows for measuring the states of its object of measurement requires formalization. Formalization implies describing an object or a phenomenon with a limited set of formal propositions only. A formal proposition is characterized by a clear truth value. While formalization offers clear advantages such as validation of propositions, the claim for formalization faces a substantial challenge: Obviously, there are aspects in the realm of organizational performance that resist formalization. These are particularly aspects based on intentional semantics, i.e. meaning is constituted by references to sensual experiences. If intentional aspects are formalized anyway, the resulting indicators may impede a differentiated appreciation of social reality that makes use of "empathy (in German *Einfühlung*) or re-creation in the mind of the scholar of the mental atmosphere, the thoughts and feelings and motivations, of the objects of his study" (von Wright, 1971, 6). This is not just an objection raised by an avid proponent of hermeneutic research. Many researchers in organizational studies emphasize that action systems cannot be reduced to "objective" descriptions or to mechanistic control structures. Weick draws a picture of organizations in clear contrast to the vision of "objective" management controls: "Organizations, despite their apparent preoccupation with facts, numbers, objectivity, concreteness, and accountability, are in fact saturated with subjectivity, abstraction, guesses, making do, invention, and arbitrariness . . . just like the rest of us." (Weick, 1980, 5). With respect to management control, Pfeffer regards "symbolic action" as more important than the use of "objective" control systems (Pfeffer, 1981, 5; similarly Weick, 1979, 20). While indicators are intended to be objective measures of reality, they are, in the end, social constructions that "create the reality as to what performance is" (Lebas and Euske, 2007, 134), which makes them a part of the perceived reality (Nørreklit et al, 2007). With respect to using indicators for control purposes, indicators are not just regarded as an analytical tool to measure performance, but as an instrument to *promote* organizational performance. While attractive incentives schemes may have a positive effect on reference objects, they can also cause dysfunctional effects by promoting opportunistic action (e.g. Ridgway, 1956; Perrin, 1998; Neely et al, 2007). But organizational performance is not only threatened by conscious opportunism. Due to their social construction, indicators may become what Habermas calls "objectified instrumental rationality" or, in German, "vergegenständlichte Zweckrationalität" (Habermas, 1968); they lose their original meaning and become an end in themselves that stakeholders regard as important for the organization. While this may lead to an ostensibly successful reproduction of organizations, it may also be a threat to long-term competitiveness. The more organizational action is determined by constructions that are oversimplifications, that do not account for relevant aspects of organizational reality, and do not sufficiently reflect relevant boundary conditions, the less its ability to cope with a changing environment.

We derive a number of implications for the design of MetricM from these considerations. Above all, it is assumed that the issues raised in this section recommend a rational-analytical approach to cope with the complexity of management in large organizations. Performance indicators are, in this respect, seen as a means of analytical support to corporate management. We further presume that a method aimed at supporting such a rational-analytical approach should encourage the reflective design and use of performance indicators and performance measurement systems and build upon rational discourses among the stakeholders involved. We do not, however, assume that to allude to epistemology, theories of truth and dysfunctional consequences suffice to promote a rational approach to performance measurement practice. Rather, we presume that the method itself should cultivate a differentiated, analytical and rational approach to the design of performance measurement systems and, at the same time, should acknowledge their multifaceted organizational consequences. The cultivation of such an approach recommends the adoption of a relaxed attitude toward performance indicators and their organizational use. In particular, it acknowledges the binding character of indicators but emphasizes that indicator systems should not be taken for granted and should always be regarded as subject to discussion and to further development. It also emphasizes to complement the "rational", "objective", indicator-based management style with an understanding of management as symbolic action as a regulating amendment. The method should, therefore, place emphasis on

1. Transparency and traceability (e.g. of intentions and assumptions) and the need for indicator justification: The *transparency* precept recommends identifying all intentions and assumptions underlying the conception and

use of indicators and their relations to other indicators to make those considerations explicit and, hence, accessible to debate and critique. It also emphasizes the need for a rationale for a performance indicator and its relations, through appropriate justification procedures. Given the complexity of performance measurement systems, allowing for and cultivating a multi-criteria justification is recommended.

2. Precision of indicator definition: The *precision* precept acknowledges the limits of formalization yet underlines the importance of precise conceptualizations. The primary objective of the design of performance measurement systems is to foster a shared understanding and a common interpretation of organizational performance and its constituents. Both—a shared understanding and a common interpretation—presuppose that the performance indicator specifications and their relations forming the performance measurement system are precise, consistent and coherent (Lebas and Euske, 2007).

3. Different perspectives of (groups of) stakeholders affected by indicators: The *multiple perspective* precept is based on the observation that performance indicator systems are relevant for and used by various groups of stakeholders from senior management to knowledge workers (Atkinson et al, 1997). Different groups of stakeholders have different demands regarding the types of indicators and of reference objects, their level of detail and of abstraction. This precept recommends accounting for the perspectives of (groups of) involved stakeholders with a particular emphasis on those affected by later use of the performance measurement system.

4. The usage context of performance indicators: The *context* precept does call for acknowledgment that indicator systems are situated in a specific organizational setting and, thus, require interpretation with reference to the organizational context in which they are embedded (Neely et al, 1997). Hence, indicators should not be specified independently, but with particular attention to their links to the organizational context they are utilized in. The precept also recommends the review and reworking of performance measurement systems on a regular basis that suggests a representation of indicators that is convenient to revise and to develop further.

These four precepts are refined in the next section to justify domain-specific requirements. Albeit, one particular aspect remains outside of the scope of the method design. The modeling method we intend to design is likely to flourish in an organizational culture that appreciates critical reflection and challenges to the status quo and that promotes a critical attitude toward indicator systems. However, the implementation and cultivation of such an organizational culture cannot be designed into a modeling method—whether usage of a method affects and possibly changes an organizational culture remains, however, an interesting question.

4.2 Requirements and key concepts

This section refines the principal design goals stated in the introductory section—cultivating reflection, stimulating communication, improving transparency and traceability—to establish five domain-specific requirements a method aimed at supporting the reflective design and use of performance indicator systems should satisfy. The requirements analysis is informed by the four precepts derived in the previous section. This section also summarizes the initial conceptual reconstruction of the technical terminology used in the performance measurement domain by identifying essential domain-specific concepts. Both the requirements and key concepts guide the development of MetricM. They also serve as a conceptual frame of reference for the initial method evaluation in Section 6.

Requirement 1 (Rationale) *A method should provide concepts that allow for a differentiated representation of the rationale underlying an indicator and its relations. It should provide the means to justify the existence and importance of (relations between) performance indicators and to reveal intentions and assumptions informing indicator justification.*

Key concepts: intention; assumption; justification (of a performance indicator and its relations).

Rationale (following precept 1). To support proper interpretation, especially by those who did not participate in the design process, and to attenuate dysfunctional consequences, it is advisable to substantiate performance indicators and their relations by virtue of a traceable rationale that assists in reflecting later use of the system by making intended interpretations explicit (Nørreklit et al, 2006, 2007). Such a rationale should include the purpose of the performance indicator and its relations, the intentions of its designers, sources of data and further assumptions underlying the choice and selection of the indicator (Neely et al, 1997, 1151).

Requirement 2 (Coherence and consistency) *A method should support—and, if possible, enforce—the design of coherent and consistent performance indicator systems. The method should, therefore, provide a precise performance indicator conceptualization and account for the precise and differentiated representation of relations among performance indicators.*

Key concepts: indicator; relations between indicators.

Rationale (following precept 2). A precise conceptualization of a performance indicator is a prerequisite to rational discourse on indicator matters (Eccles and Pyburn, 1992), and to analyses of indicator systems (Popova and Sharpanskykh, 2010). Likewise, relations among indicators and between indicators and other relevant concepts require a precise conceptualization. Indicator systems that lack precision or important aspects or are partially inconsistent or even incoherent jeopardize their very purpose (Ridgway, 1956; Perrin, 1998).

Requirement 3 (Multiple perspectives and levels of abstraction) *A method should provide meaningful representations of indicator systems on various levels of abstraction to satisfy the needs of affected groups of prospective users. To foster an intuitive use of the method, each perspective should, as far as possible, correspond with the abstractions, concepts and (visual) representations known and meaningful to the targeted group of stakeholders. All perspectives should, on the other hand, be integrated with each other to foster cross-perspective communication and cooperation.*

Key concepts: perspective, organizational role, organizational responsibility.

Rationale (following precept 3). Performance measurement as a group process involves stakeholders with different professional backgrounds and responsibilities as well as specific sentiments about performance indicators and their effects (Neely et al, 1995; Bourne et al, 2000; Tuomela, 2005). Therefore, a method in support of the reflective design and use of performance measurement systems needs to take the perspectives of stakeholders with different professional backgrounds—from senior management to IT operations—into account.

Requirement 4 (Organizational context) *A method should account for the relevant organizational context and, thus, allow for the differentiated representation of relations between performance indicators and the surrounding action system composed of all organizational entities relevant to their proper interpretation.*

Key concepts: organizational context (through reference objects to the organizational action system); indicator-context relations.

Rationale (following precept 4). The organizational context in which a performance indicator is designed to be used is of particular importance to its accurate interpretation (Neely et al, 1997). The organizational context would include, for example, the organizational action system, its institutions and actors, their roles, responsibilities and corresponding decision and analysis scenarios. Hence, a method in support of the design and use of performance measurement systems has to account for the concepts representing the action system such as resources and processes the indicator relates to, and the organizational roles involved in its use (Neely et al, 1995; Bourne et al, 2000; Tuomela, 2005).

Requirement 5 (Organizational goal) *A method should allow for associating an indicator with the organizational goal the indicator is designed to represent. It should provide means to make the intended indicator-goal relation explicit and should account for a differentiated representation of indicator-goal relations.*

Key concepts: goal; indicator-goal relations.

Rationale (following precept 4). Performance indicators are *surrogates* of performance and, hence, means to observe and indicate the achievement of set objectives of an organization—not an end in themselves (Lebas and Euske, 2007). Consequently, accurate interpretation of a performance indicator is fostered by linking it to the organizational goal(s) it is designed to represent (Neely et al, 1995, 1997). Note that this requirement further refines Req. 4 (Organizational context).

5 Method design

5.1 Conceptual foundation

The research on METRICM builds on the "Multi-Perspective Enterprise Modeling" (MEMO) method (Frank, 1994, 2002). The rationale for choosing MEMO over, e.g., ARIS (Scheer, 1992, 2000) or ArchiMate (Lankhorst, 2009) is based on several considerations: (1) MEMO provides an extensive set of modeling constructs relevant to modeling performance indicators, e.g., for the modeling of organizational units, roles, resources, and IT artifacts; (2) in contrast to proprietary approaches, the specifications of the MEMO method—especially its meta models—are publicly available and documented in several publications; and (3) MEMO is based on a language architecture extensible through domain-specific modeling languages (Frank, 2008). In MEMO, domain-specific modeling languages are specified using the MEMO Meta Modeling Language (MEMO MML) (defined at the meta-meta or M_3 level). Using MEMO MML for defining and reusing common concepts at the meta level (M_2) leads to integrated models at type level (M_1), e.g., an organization structure model integrated with a business process model, a model of an IT landscape, and a performance indicator model.

Thus, the MEMO family of modeling languages promises a number of advantages for the development of the domain-specific modeling language, METRICML. Each modeling language in MEMO provides a set of reusable modeling concepts for the aspects they focus on. Of particular importance for indicator modeling are, for instance, (1) concepts for modeling organization structures (to assess indicator scope and responsibilities); (2) business processes, services, and organizational resources (to determine the action system surrounding an indicator); and (3) organizational goals and objectives (to analyze indicators with respect to strategy). In this regard, the strategy modeling language (MEMO SML) provides concepts such as "strategy" and "goal" and offers "strategy nets" and "value chains" as diagram types (Frank and Lange, 2007); the organization modeling language (MEMO ORGML) provides concepts for modeling business processes and organizational structures, e.g., "process", "event", "organizational unit" (Frank, 2002, 2010b); and the resource and the IT modeling languages (MEMO RESML and MEMO ITML) allow for modeling organizational resources in general (e.g., "human resource"), IT resources in particular (e.g., "hardware", "software", "information system"), their relationships to each other

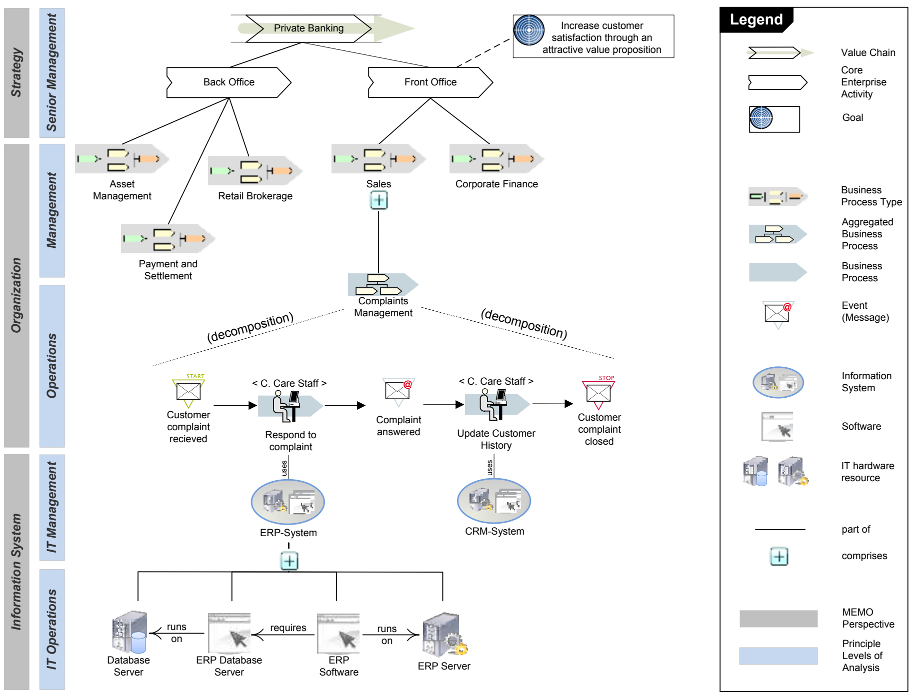

Figure 1: Key notation elements, principal levels of analysis, and elementary perspectives in the MEMO method.

(such as "uses" or "comprises") and to the business processes they are used in (Frank et al, 2009).

In principle, MetricM can be adapted to any enterprise modeling approach. Thus, MEMO serves as a typical representative of enterprise modeling methods in the context of the development of MetricM. Figure 1 shows key notation elements and principal levels of analysis supported by the MEMO method and family of modeling languages. It includes an IT resource model at the level of IT operations, a business process model showing an aggregated process ("Complaints Management") and its disaggregated control flow at the level of business operations as well as a service ("Customer Care") and a business process type ("Sales") at the level of operations management. A value chain model and an associated business objective ("Goal") at the senior management level are also shown. These five levels of analysis refine the three elementary perspectives provided by the MEMO method, i.e. strategy, organization, and information systems, and represent *typical* levels of analysis that users of the MEMO method adapt to fit their problem space.

5.2 Language design, procedural guidelines and design rationale

This section describes the MetricML specification by introducing key concepts of the language and their semantics, and by discussing essential design decisions. Procedural guidelines on the application of language concepts—as part of the process model—are introduced along with the language description.

The abstract syntax of MetricML is specified as a meta model and shown in Figure 2. Its concrete syntax, the corresponding graphical notation, is depicted in Figure 3. Four constituents are central to the language specification: (1) the *Indicator* language concept and the ancillary concepts *RationaleSpec* and *IndicatorCategory*; (2) the two inter-indicator relations: *IndicatesRelation* and *TransformsRelation*; (3) the four indicator-context relations: *refers to* (*ReferenceObject*), *is accountable for*, *acts on*, and *measures*; and (4) the indicator-goal *RepresentsRelation*. Further MetricML concepts not discussed in this section include the inter-indicator relation *is of similar kind* and the metatype *IndicatorAttribute* (both described in Frank et al, 2008) as well as the organizational context concept *DecisionScenario* currently under development as part of a major revision of the MEMO Organisation Modelling Language (Frank, 2010b).

Ad (1) The *Indicator* language concept. The *Indicator* concept serves to describe the essential characteristics of a performance indicator type. The approach taken here conceptualizes performance indicator as a dedicated language concept, the MetricML *Indicator* metatype, to allow for modeling its relations to other indicator types, to reference object types representing organizational context and to goal types. An alternative "attribute" approach conceptualizes performance indicator as (meta-) attribute of metatypes (e.g. "average through-

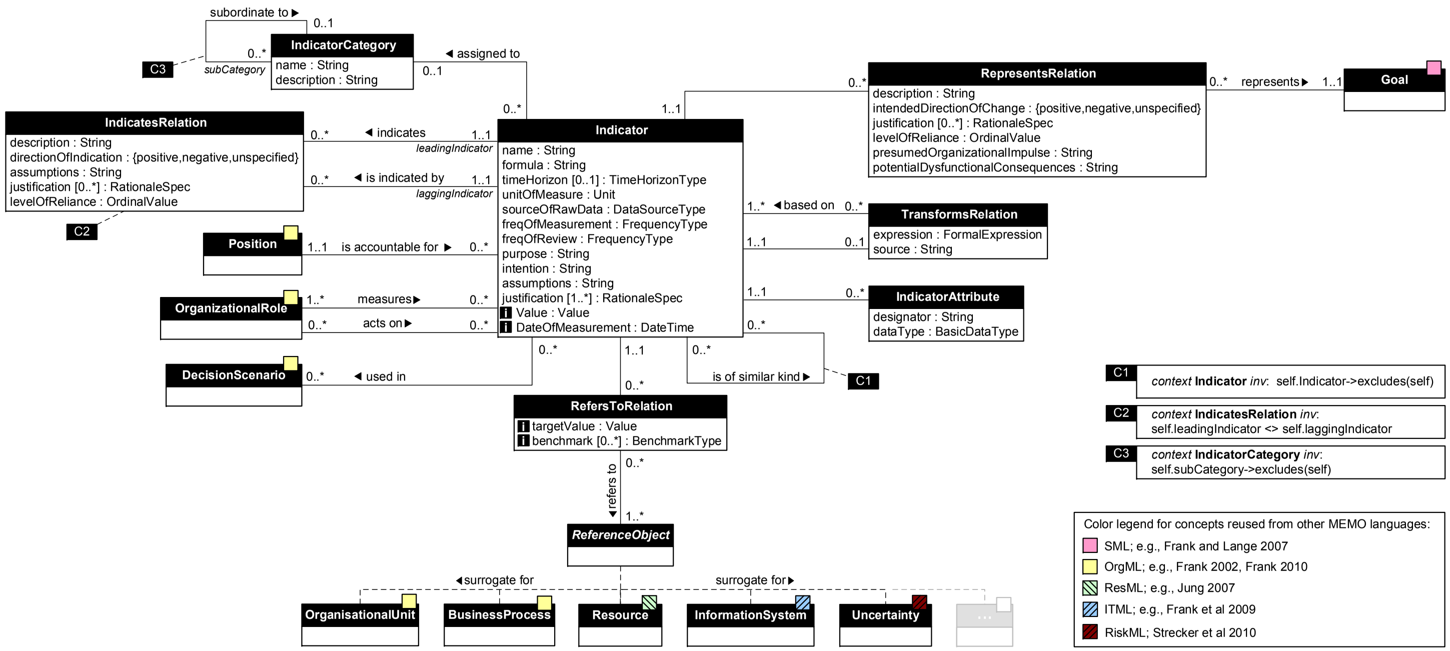

Figure 2: The MetricML metamodel (language specification) in the MEMO Meta Modelling Language (Frank, 2008).

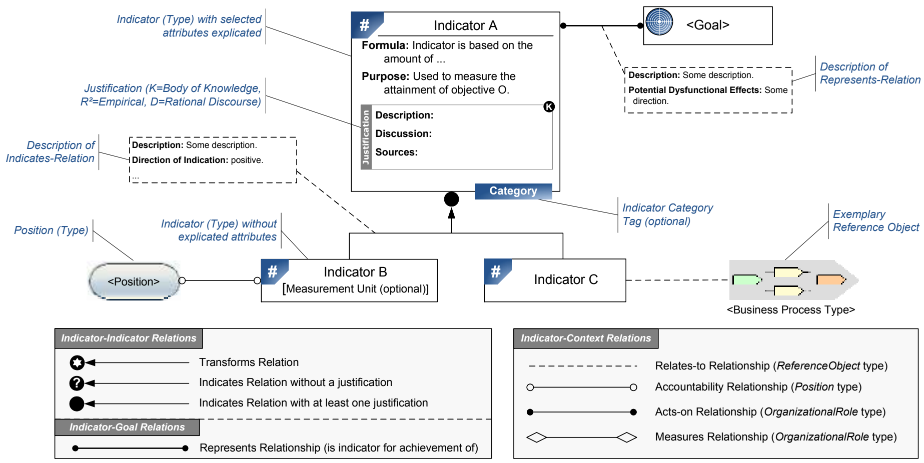

Figure 3: Overview of key notation elements of METRICML refined from (refined from Frank et al, 2008, 2009).

put time" of a business process type or "average number of employees" of an organizational unit type). However, such a conceptualization does not allow for representing inter-indicator relations and, thus, is of limited use in the light of the requirements discussed in Section 4.2.

It is anticipated that an indicator type may be defined at a wide range of different organizational levels of abstraction from high-level financial indicators (e.g. "sales turnover") to low-level operations indicators (e.g. "time to quote"). The case studies reconstructed from Neely et al (1997) depicted in Figure 4 include "Sales turnover", "Time to quote", and "Customer service—Adherence to customer schedule". Note how the attributes of each indicator type are specified. For instance, the "Sales turnover" indicator type is deliberately specified to allow for associating it with different reference object types for which planned sales are defined and invoiced sales records are available. The modeler, thus, may associate the "Sales turnover" indicator type with a "Sales" business process type, a "Division Healthcare" organizational unit type, and a "Cosmetics Products" product family type. The *Indicator* concept also provides the flexibility to allow for another approach to indicator type definition if sufficient differences warrant a separate type definition. If, for example, the frequency of measurement of the sales turnover of the "Sales" business process type should be different from the frequency of measurement of the sales turnover of the "Division Healthcare" organizational unit type, it is recommended to create two different indicator types, e.g. "Sales turnover—Sales process" and "Sales turnover—Division Healthcare". This flexibility, however, does increase the cognitive load for the modeler to ensure consistent indicator models. We chose this particular language design over more restrictive—and, possibly, less demanding—designs, because an indicator type specification primarily depends on the level of abstraction at which reference object types are defined and on the specific intended purposes and analyses. Both cannot be anticipated at language design time.

Neely et al (1997) review pertinent literature on performance indicator definitions to synthesize a “performance indicator record sheet”, a textual template describing essential characteristics of indicators based on requirements identified in literature. They refine their initial template in case studies to arrive at a modified record sheet. Its indicator definition contains descriptors similarly found in the conceptualizations by Popova and Sharpanskykh (2010) (i.e. Scale, Source, Time Frame, Owner) and by Pourshahid et al (2007) (i.e. targetValue, kpiValueDataSource, kpiReportDataSource). The MetricML *Indicator* concept refines these conceptualizations. A MetricML indicator type is described by a *formula* (a brief verbal description of how the indicator value is calculated), a *UnitOfMeasurement* (can be “none” or “index” in the case of non-dimensional measures such as customer satisfaction; or any specific unit of measurement, e.g. “Percent”, “Minutes” or “US dollars”) and a *TimeHorizon* complementing the measurement unit (can be, for instance, “week”, “month”, “year” etc. to form a measurement dimension such as “Percent per Year”), its *sourceOfRawData* (the data source from which the indicator is calculated, either as a textual description, e.g. “invoiced sales records”, or as a fully qualified data manipulation language statement, e.g. “SELECT FROM . . . ”) and the frequency of measurement, *freqOfMeasurement* (how often the measurement will be taken). In the metamodel, the two attributes *Value* and *DateOfMeasurement* are marked with an 'i' printed as white on black, characterizing them as “intrinsic attributes” (Frank, 2008). These attributes reflect a characteristic that applies at the instance level, although in the language specification it is associated with types. Hence, an intrinsic attribute is *not* instantiated at the type level but at the instance level, thereby enabling representation of the current value and its date of measurement of a concrete indicator instance.

The reflective definition of a performance indicator is maintained by describing its *purpose* (a verbal statement by the indicator designers), the (managerial) *intention* underlying the indicator definition, the *assumptions* associated with the intended indicator use as well as the rationale that justifies indicator design and use *(justification)*. By making these considerations part of each indicator definition, it is intended to foster its critical review, to stimulate management discussions on indicator (mis-) use, and to make the results of these discussions a persistent part of the resulting (conceptual model of the) performance measurement system. As illustrated by the examples in Figure 4, *purpose*, *intention* and *assumptions* need to tend to a wide range of different writing and discussion styles. Thus, the “String” data type. This is different for *justification* for which we propose to differentiate three idealized justification procedures (see Section 4.1) as means to remind discussants of the importance of substantiating each performance indicator and each of its relations and that the complementary use of these justification procedures strengthens their rationale. In this context, the main challenge is to provide a specification that supports the user in applying different justification procedures. The attribute type *RationaleSpec* is introduced to provide such a specification (see Figure 5): It allows the language

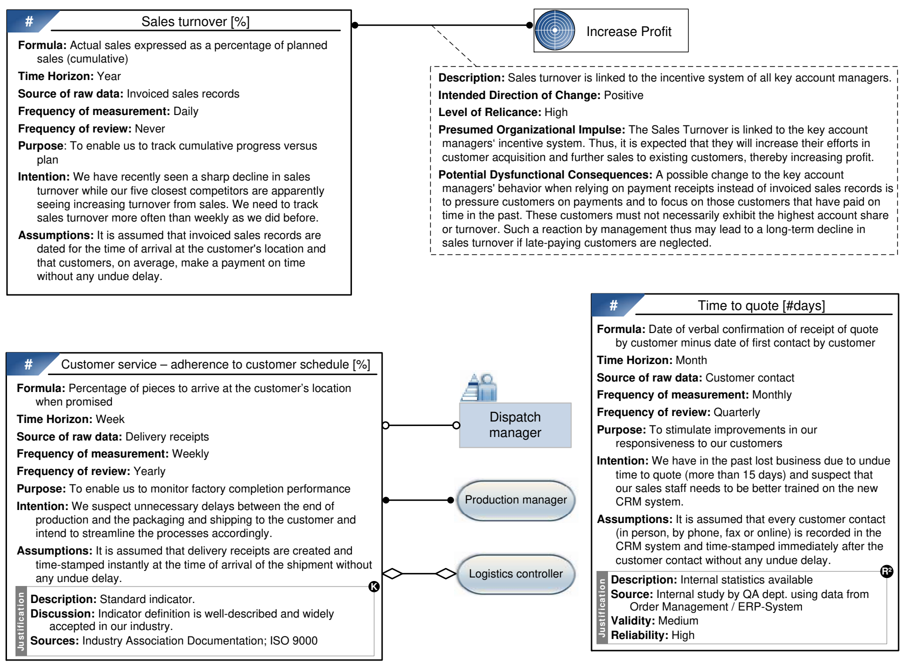

Figure 4: Reconstruction of three indicator specifications based on case studies described in Neely et al (1997) with extensions for reflective design added for illustration purposes.

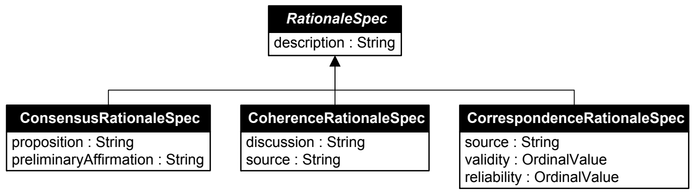

Figure 5: The *RationaleSpec* concept.

user to specify *propositions* and a *preliminaryAffirmation* to describe the results of rational discourse, *discussion* and *sources* to clarify agreement and conflict with an existing body of knowledge, and *sources*, *validity*, and *reliability* to represent empirical evidence as part of the rationale.

As a further refinement, the *IndicatorCategory* metatype is introduced as a way of structuring large sets of indicator types according to user-defined criteria. Since *IndicatorCategory* types can form a hierarchy, it is possible to apply this language concept to assign an instance of *Indicator* to hierarchical organizational levels (e.g. "financial", "operations", "IT") or to other abstractions deviating from regular organizational structures (e.g. "senior management", "middle management", "operations management"). Hence, the *IndicatorCategory* concept also provides a means to reuse existing indicator categorizations associated with performance measurement frameworks such as the Balanced Scorecard (i.e. "Financial", "Customer", "Internal Business Process", "Learning & Growth") and to reproduce corresponding visualizations if needed.

Ad (2) The inter-indicator relations. Performance measurement literature generally conceptualizes relationships between performance indicators as a "cause-and-effect relationship" (Kaplan and Norton, 1996a, 31) and, hence, as a "causal model" (Lebas and Euske, 2007, 127). However, as shown, for example, by Nørreklit (2000) and Nørreklit (2003, 616–617), in this conceptualization, causality is misconceived and the causality assumption is invalid which may lead to "the anticipation of performance indicators which are faulty, thus resulting in dysfunctional organizational behaviour and sub-optimized performance (De Haas and Kleingeld, 1999)" (Nørreklit, 2000, 75). Giving up on the causality claim, however, does not mean that relationships between performance indicators cannot be conceptualized at all. Rather, as suggested e.g. by Nørreklit (2000, 83) and Malina et al (2007), relationships among performance indicators can either take on a logical or an empirical form.

The mathematical transformations known from the *DuPont system of financial control* are examples of logical relationships. In fact, their relationship is dependent on definition (of ROI) and is, thus, a tautology. (Tauto-)Logical relations between indicators are specified using the *TransformsRelation*. The mathematical transformation of one or between two or more indicators forms the resulting indicator. For instance, in the DuPont system, the return on investment may result from multiplying net profit margin by total assets turnover.

The *TransformsRelation* is, thus, specified by an *Expression* and a reference to the literature in which the logical relation is specified and documented (*Source*). It is recommended to establish a standard notation in which all expressions in the scope of the performance measurement system are formulated. With regard to the work by Popova and Sharpanskykh (2010), such a standard could use a formal logic if the intention is to support partly automated validation of the indicator system.

Empirical relationships can be based on past experiences including statistical analyses of data generated within the firm (in an inductive fashion), or on results of empirical studies conducted outside of the firm. Starting from these considerations, the empirical relations among performance indicators are conceptualized as a deliberately *simplifying* "is assumed to indicate" interpretation in which ("leading") performance indicators are iteratively perceived as indicators to *possible* changes of subsequent ("lagging") indicators at increasingly higher organizational levels of abstraction to form a directed acyclic graph—a (poly-) hierarchy—of indicators. This conceptualization is for analytic reasons and shall be seen as a consciously deployed means of reducing complexity. It is important to complete this truncated interpretation: Performance indicators are surrogates of performance (Lebas and Euske, 2007), i.e. they have to be interpreted as indicators of observable and measurable outcomes of actions carried out to pursue a business objective subsequent to making a decision:

> *"The key concept is not that the metrics themselves must have a direct causal effect on eventual outcomes (or other macro goals). The key concept is that the metrics are chosen so that actions and decisions which move the metrics in the desired direction also move the firm's desired outcomes in the same direction"* (Hauser and Katz, 1998, 520).

In performance measurement literature, a certain time-lag is commonly assumed between the realization of measurable outcomes (for a discussion, see Nørreklit, 2000), so that indicators at higher levels of for-profit organizations typically relate to financial outcome variables such as sales turnover or return on capital employed (assumed to require a longer period of time to show), whereas intermediate indicators include customer satisfaction or service quality (assumed to require less time to show), and lower-level indicators include process throughput or number of customer complaints (assumed to require an even shorter period of time to show) (e.g. Neely et al, 2000). The conceptualization as a "is assumed to indicate" relationship, hence, denotes a conscious simplification of complex interactions for analytical purposes. It should be read as "Indicator A_i measures an outcome of an action A_a following a decision A_d. The outcome of action A_a is likely to have an influence on (decision B_d preceding) an action B_a (either internal to the firm, e.g. a managerial decision and an operational action, or external to the firm, e.g. a buying decision and a purchase by a customer). As a result of a deliberate design process of the performance measurement system, a change of the value of Indicator A_i is assumed to indicate a (likely) change of the value of Indicator B_i measuring the outcome of action B_a." Hence, the reasoning and justification behind an instance of this relationship depends on specific situational factors such as the application domain or the purpose of the performance measurement system. It is therefore strongly recommended to provide at least one *justification* for the *IndicatesRelation*, to make the assumptions underlying the *IndicatesRelation* explicit and to assess the designers' *LevelOfReliance* with respect to the modeled relation. Figure 6 illustrates the prototypical application of the inter-indicator relation types, *TransformsRelation* and *IndicatesRelation*, based on a case study of the performance measurement system of the National Insurance Company (Kaplan and Norton, 1996b, 73–77).

Ad (3) The indicator-context relations. The integration of MetricML with the MEMO family of modeling languages permits enrichment of the description of indicator types with relevant organizational context. Due to the metalevel integration, metatypes deemed as relevant reference objects for indicator types can be reused from other MEMO languages (indicated by the colored rectangles in the concepts' header in Figure 2) to associate an indicator type with the reference object(s) it refers to. The *ReferenceObject* concept thus functions as a surrogate for *BusinessProcess* and related types (e.g. *OrganizationalUnit* and *Resource*). Hence, organizational context in MetricML is not restricted to a metatype for processes as in related work but involves a set of differentiated metatypes describing the surrounding action system, i.e. the organizational setting in which an indicator type is to be interpreted.

Proper interpretation of an indicator in its organizational setting benefits from a reference point such as a benchmark or target value (Neely et al, 1997). Meaningful interpretation of such a reference point presupposes an association with the respective reference object the benchmark or target value is directed at. Unlike in earlier work (e.g. Popova and Sharpanskykh, 2010; Pourshahid et al, 2007, 2008), both *targetValue* and *benchmark* therefore qualify the *RefersToRelation* between an indicator type and reference object type and not the *Indicator* concept itself. This provides language users with additional flexibility. On the one hand, it allows the definition of different target or benchmark values for an indicator for the (different) reference object(s) it refers to. On the other hand, MetricML supports defining the same reference points, i.e. the same *RefersToRelation*, for different reference objects. To a certain degree, the semantics of the *RefersToRelation* depends on the level of abstraction at which reference object types are defined (as well as on the intended purposes and analyses of the indicator model). However, the actual values of both *targetValue* and *benchmark* are likely to change frequently—in most cases at least every year. Thus, both *targetValue* and *benchmark* are specified as intrinsic attributes. They are not instantiated at type level but at instance level to prepare for the implementation of indicator-based information systems such as management dashboards.

The integration with other MEMO modeling languages enables domain-specific analyses, for example, the identification of (potential) dependencies between indicator types as illustrated in Figure 7. It depicts two indicator

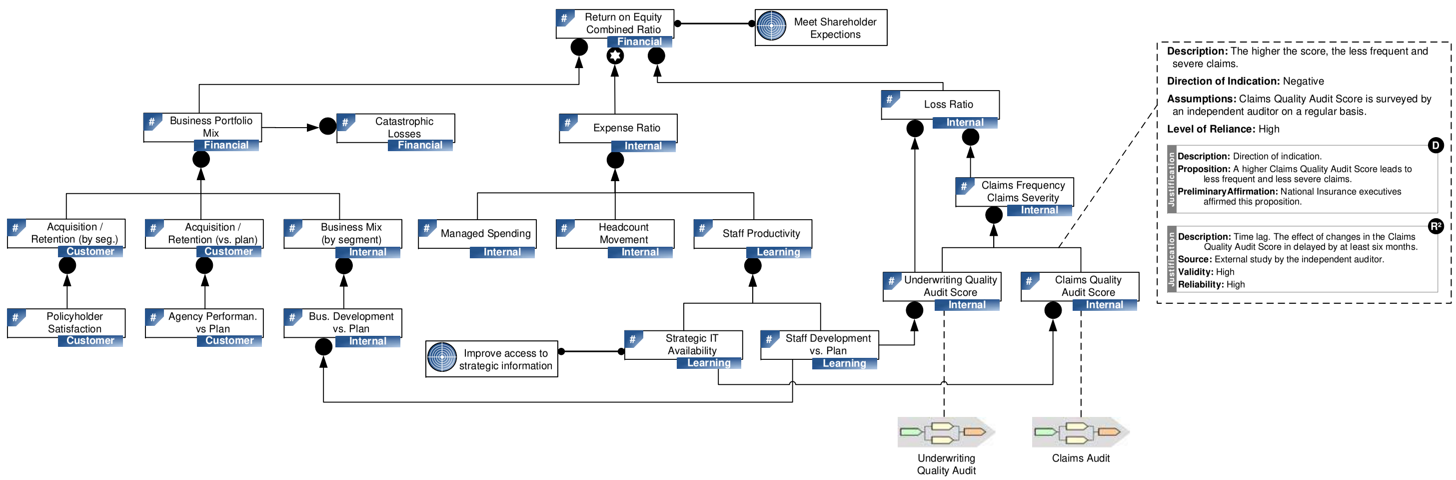

Figure 6: Reconstruction of the National Insurance case study (Kaplan and Norton, 1996b, 73–77) in METRICML with extensions for the reflective design and use of the performance measurement system added.

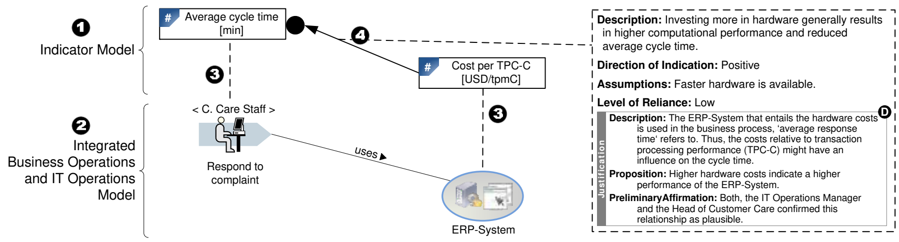

Figure 7: Identifying important yet overlooked dependencies between indicators based on the analysis of indicator-context relations.

types that are specified independently of each other (see Step 1 in Figure 7). The integration of the indicator model with an enterprise model (Step 2) permits association of the indicator types with their respective reference objects (Step 3). Thus, the analyst can check for and assess *potential* yet unidentified (i.e. "hidden") relations between indicators. For this purpose, the language user can trace the associations among reference objects in the enterprise model. For instance, the indicator type measuring a specific characteristic of a resource (e.g. "US dollar per TCP-C") is likely to relate to indicator types measuring the processes the resource is allocated to ("Average cycle time"). As a result of the visual inspection, the modeler is able to enrich the indicator system by identifying this relation between these two indicators and by making it explicit, i.e. by associating the two indicator types via an *indicates* relationship (see Step 4). If a timely execution of the dependent process is more important than the relative costs of this resource, this additional relation, i.e. the integration with the organizational context, contributes to a better understanding of—formerly hidden—dependencies.

Further organizational context is provided through the differentiated representation of the involvement of different groups of stakeholders. For instance, Neely et al (1997, 1140) suggest two types of involvements of *organizational roles* in indicator use: The one collecting and reporting an indicator (*measures*) and the other one acting on the occurrence of a certain indicator *value* (*acts on*). In the light of present auditing standards such as COBIT, an organizational unit, i.e. a specific *position*, *is accountable for* an indicator. In principle, two design alternatives are feasible to represent these types of involvement. First, for each identified type of involvement a particular association between the metatypes representing the organizational units and the indicator is established. Second, a metatype serves as an "association class" between the organizational units concepts and the indicator concept and allows us to instantiate these three—and further—associations. While the latter alternative provides more flexibility for modelers to add company-specific relations (e.g. "informed") without adapting the metamodel, the first alternative restricts modelers to predefined types of involvement and their min/max-cardinalities—and is thus likely to promote a more secure modeling. A prototypical application of these relations is illustrated in Figure 4 (indicator type “Customer service—adherence to customer schedule”).

Ad (4) The indicator-goal relation. The *RepresentsRelation* serves to specify the relation between a goal type and an indicator type. A prototypical application of the language concept is provided in Figure 4 (indicator type ‘Sales turnover’ and goal type ‘Increase Return on Investment’). The explicit differentiation between goal type and indicator type promotes separation of concerns. ‘Performance indicator’ and ‘business objective’ represent two abstractions aimed at different purposes even though their boundaries often seem to blur in management practice (Kaplan and Norton, 2004). Moreover, stakeholders involved in setting business objectives may not be involved in designing the corresponding performance measurement system and vice versa. Having goal types and indicator types as separate language concepts, thus, facilitates division of labor and acknowledges distribution of expertise in an organization. Differentiation at type level necessitates, however, making explicit links between indicator types and goal types to support indicator reflection and interpretation. Specifically, the relation between an indicator type and a goal type embodies the hypothesis on how changes to the indicator value affect goal achievement, or put it another way—how the goal is represented by the indicator (see Section 4). The conceptualization of the *RepresentsRelation* therefore distills this hypothesis including the intended direction of change (how does a change in the indicator value relate to a change in goal achievement?) and the presumed organizational impulse of the linkage (how should use of the indicator affect organizational behavior?). Potential dysfunctional consequences of indicator use with respect to goal achievement *(potentialDysfunctionalConsequences)* should be (re-)considered as part of the design process of the performance measurement system and explicitly annotated in the *RepresentsRelation* as well as the level of reliance of the hypothesis stated and the behavioral consequences (decisions and actions) intended by the system designers *(intendedOrganizationalImpulse)*. Again, a rationale should be provided at the time of design to justify the hypothesized relationship between an indicator and an organizational goal *(justification)*.

The resulting visualization of an indicator with reference to the organizational goal it represents contributes to coherent performance measurement systems by highlighting its usage context and by advancing subsequent discourse. In general, it is therefore recommended to associate an indicator type with the goal types it is directed at. Whether an indicator type *can* be associated with a goal type depends on level of detail and granularity of the goal system. In many cases, organizational goals will be defined for high-level objectives, so that the linkage between an indicator type and the goal type is too remote and distant to be justified. In such cases, the *IndicatesRelation* should be used to express the indirect relation between the indicator type and the goal type via other indicator types. Figure 8 illustrates the use of both direct and indirect indicator-goal relations. In some cases, modeling indicator-goal relations will reveal gaps in

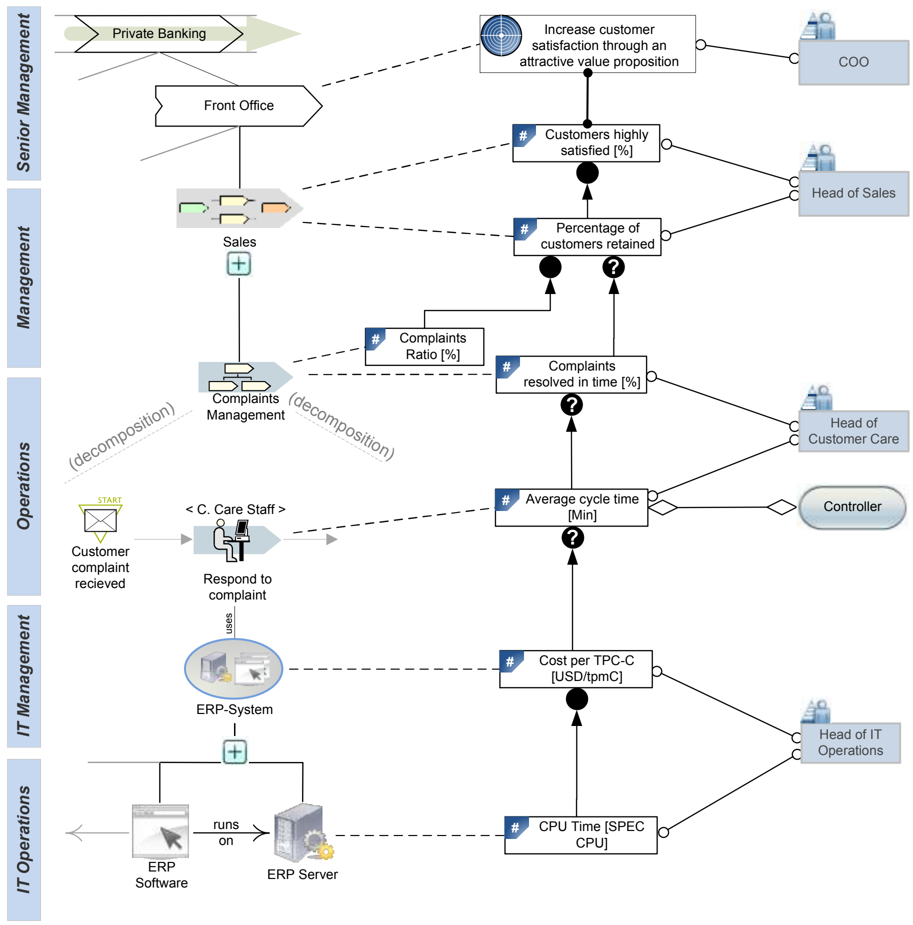

Figure 8: Integration of enterprise models with a model of a performance measurement system as a basis for further analyses at different organizational levels of abstraction—principal perspectives and levels of analysis shown at far left (compare with Figure 1).

the goal system and inspire the formulation of new goal types which, in turn, can be associated with indicator types. For example in Figure 8, interpretation of the indicator type 'Average response time' could benefit from its association with a goal type 'Increase internal user satisfaction through responsive application systems' or, more specifically, 'No more than 10 percent of user requests with a latency of 2 seconds or more'. Furthermore, the figure also provides a prototypical application of how MetricML supports different perspectives. It extends the diagram shown in Figure 1 by selected indicators that measure specific reference objects on various organizational levels from IT operations over business operations up to the senior management level. The identification and analysis of indicator-goal and inter-indicator relations at the various organizational levels contributes to stakeholder comprehension and facilitates discourse on indicator matters.

6 Method evaluation

The three primary design goals stated in Section 1, cultivating the reflective design and use of performance measurement systems, improving transparency and traceability of indicator matters, and stimulating communication among involved stakeholders, are refined and operationalized through the five domain-specific requirements. The method evaluation is based on the assumption that by satisfying these requirements, the design goals are met. In turn, this implies that hypotheses underlie the relationships between high-level design goals and domain-specific requirements. It is, for example, assumed that if a method in support of the reflective design and use of performance measurement systems prescribes documenting managerial intentions and assumptions, justifying indicators and their relations, and prescribes reflecting upon potential dysfunctional consequences, rational discourse is encouraged; and that traceable interpretation of performance indicators is facilitated. In other words, the method positively contributes to approaching the primary design goals. The remaining hypotheses are interpreted accordingly but not made explicit, since the presumed effect and its direction can be inferred from Section 4.

Table 2 links method features to requirements and evaluates if and how MetricM addresses the five requirements. Table 3 refines the method evaluation by showing whether and how MetricML language concepts account for key domain-specific concepts (see Section 4.2) and by contrasting MetricML with prior work (see Section 2). Generally, MetricM attends to each requirement by both dedicated modeling concepts and corresponding procedural guidelines. MetricML (re-)constructs the semantics of domain-specific concepts, reconceptualizes modeling concepts proposed in related work (e.g. *TransformsRelation*) and introduces advanced modeling concepts (e.g. *RationaleSpec*). The MetricM method specifically addresses the requirements R1 *Rationale* and R2 *Coherence & Consistency* through dedicated concepts (e.g. *justification*) and guidelines. In particular, MetricML frames a space of potential extensions to current business practice by introducing dedicated modeling concepts

Table 1: Evaluating MetricM: Method features contrasted with domain-specific requirements.

Requirement	Description	MetricM features and discussion
R1 *Rationale*	A method should provide concepts that allow for a differentiated representation of the rationale underlying an indicator and its relations. It should provide the means to justify the existence and importance of (relations between) performance indicators and to reveal intentions and assumptions informing indicator justification.	Dedicated language concepts and corresponding procedural recommendations contribute to substantiation of indicators and their relations. *RationaleSpec* metatype and *justification* attributes foster multi-criteria justification; *purpose*, *intention*, *assumptions* substantiate indicator type definitions.
R2 *Coherence & Consistency*	A method should support—and, if possible, enforce—the design of coherent and consistent performance indicator systems. The method should, therefore, provide a precise performance indicator conceptualization and account for the precise and differentiated representation of relations among performance indicators.	Coherence fostered by explicit consideration for organizational goal and for organizational context to facilitate indicator interpretation within its organizational setting (with semantically rich description of context through respective language concepts); Consistency fostered by precise indicator type definition and differentiated relations between indicators; Metamodel-based approach supports modeling tool development to prevent syntactical errors and to foster convenient and efficient modeling.

Table 2: Evaluating MetricM: Method features contrasted with domain-specific requirements - Part II.

Requirement	Description	MetricM features and discussion
R3 *Multiple perspectives and levels of abstraction*	A method should provide meaningful representations of indicator systems on various levels of abstraction to satisfy the needs of affected groups of prospective users. To foster an intuitive use of the method, each perspective should, as far as possible, correspond with the abstractions, concepts and (visual) representations known and meaningful to the targeted group of stakeholders. All perspectives should, on the other hand, be integrated with each other to foster cross-perspective communication and cooperation.	Reuse of MEMO perspectives and structuring concepts (e.g. *AggregatedProcess* and *OrganizationalUnit*) to tailor perspectives to the needs of stakeholders and to provide representations akin to known visualizations; Integration through common meta- and meta-metamodels. *IndicatorCategory* to visually tag and classify indicators according to user-defined typology.
R4 *Organizational Context*	A method should account for the relevant organizational context and, thus, allow for the differentiated representation of relations between performance indicators and the surrounding action system composed of all organizational entities relevant to their proper interpretation.	Surrogate *ReferenceObject* to reuse concepts provided by MEMO languages representing the relevant organizational action system, e.g. *BusinessProcess*, *OrganizationalUnit*, *InformationSystems*, or *Uncertainty*.
R5 *Organizational Goal*	A method should allow for associating an indicator with the organizational goal the indicator is designed to represent. It should provide means to make the intended indicator-goal relation explicit and should account for a differentiated representation of indicator-goal relations.	Explicit differentiation between goal type and indicator type; *RepresentsRelation* to associate indicator type with goal type and to specify further semantics of the relation (e.g. *intendedOrganizationalImpulse*, *potentialDysfunctionalConsequences*).

supporting the reflective design and use of performance measurement systems (e.g. the *IndicatesRelation* and the *RationaleSpec* respectively *justification* attribute). Requirement 3 *Multiple perspectives and levels of abstraction* is addressed by the MetricML *IndicatorCategory* metatype and through the reuse of existing MEMO concepts—including the conceptualization of a "perspective" as a cognitive predisposition (Frank, 1994, 164). MetricM, furthermore, benefits from the reuse of MEMO modeling concepts and corresponding process models with regard to both requirements R4 *Organizational context* (e.g. *OrganizationalUnit*) and R5 *Organizational goal* (e.g. *Goal*). Though, MetricML provides differentiated relations between indicator types and organizational context concepts (e.g. *RepresentsRelation*). MetricML also extends earlier work by incorporating further concepts representing the organizational action system an indicator refers to—beyond business processes, agents, and their roles. In summary, the initial method evaluation indicates that both the MetricML language concepts and the corresponding procedural guidelines of MetricM promise to appropriately support the intended design goals of stimulating communication among stakeholders, of improving transparency and traceability of indicator matters, and, lastly, of cultivating the reflective design and use of performance measurement systems.

The present design of MetricM acknowledges prior contributions e.g. to automated indicator reasoning and to the development of indicator-based software systems such as management dashboards. MetricM is thus designed to complement other approaches and to open paths to method integration: Specifically, the approach by Popova and Sharpanskykh (2010) targets automated reasoning and provides complementary concepts for expressing indicators as logic statements. Moreover, the approach by Pourshahid et al (2007, 2008) as well as our own prior work (Frank et al, 2009) is primarily aimed at the development of corresponding software systems and, among others, supplements differentiated concepts for indicator values (target, benchmark, threshold etc.). MetricM does currently not address those aspects as such but provides means for integrating with related work through dedicated modeling concepts aimed at method integration (i.e. the *IndicatorAttribute* concept and the *expression* attribute in the *TransformsRelation*). To support the development of indicator-based information systems, MetricM would require such a method integration and entail further refinements of the semantics of modeling concepts. Presently, concepts like *RationaleSpec* and *intention* rely on String data types and do not provide further semantics; hence, impeding advanced tool-supported analyses.

However, the present conceptualizations in MetricML assume that the precise semantics of concepts defined by their attributes and relations in the metamodel promotes comprehension by language users. Further research has to study the effects of the presented conceptualizations on prospective users, i.e., whether the proposed concepts indeed improve communication and cooperation or whether their graphical representations are accepted by the stakeholders involved in performance measurement system design and use. The number of concepts and the diversity of relations in MetricML suggests that the language itself may increase complexity when using the method. It is thus currently not

Table 3: Evaluating MetricML: Language concepts contrasted with domain-specific requirements, concepts and prior work. Concepts marked in *Italics* extend prior work on conceptual modeling of performance measurement systems.

Req.	Key concepts	Pourshahid et al (2007, 2008)	Popova & Sharpanskykh (2010)	MetricML
R1	Intention	–	–	*intention* (as part of *Indicator*)
R1	Assumption	–	–	*assumptions* (as part of Indicator and *IndicatesRelation*)
R1	Justification	–	–	*RationaleSpec*, *justification* (as part of Indicator and *IndicatesRelation* and *RepresentsRelation*)
R2	Indicator	targetValue, thresholdValue, worstValue, kpiValueDatasource, kpiReportDatasource	Name, Definition, Type (cont. or discrete), Scale, Min value, Max value, Source, Owner, Threshold, Hardness, PI expression	*Indicator:* name, formula, timeHorizon, unitOfMeasurement, sourceOfRawData, value *freqOfMeasurement*, *freqOfReview*, *dateOfMeasurement*
R2	Inter-indicator relations	– (indirect via Performance Model)	is_defined_over, is_based_on, causing, aggregation_of, correlated	*IndicatesRelation*, TransformsRelation, isOfSimilarKind
R3	Perspective	–	implicit "performance view"	adapted MEMO perspectives**
R3	Organizational role	–	agent, role	*OrganizationalRole**, *Position**, *DecisionScenario**
R3	Organizational responsibility	– (indirect relation via PerformanceModel)	uses, has_owner	*usedIn (DecisionScenario*)*, *isAccountableFor (Position*)*, *measures*, *actsOn (OrganizationalRole*)*
R4	Organizational context	business process	business process, environmental characteristics	*ReferenceObject*
R4	Indicator-context relation	– (indirect relation via PerformanceModel)	env_influence_on	*RefersToRelation*
R5	Goal	Performance Goal, IntentionalElement	Goal Pattern, Goal	Goal***
R5	Indicator-goal relations	Unnamed relation between KPI and Performance Goal	measures	*RepresentsRelation*

* Specified as part of the MEMO OrgML (Frank, 2002, 2010b).

** Specified as part of the MEMO method (Frank, 1994, 2002).

*** Specified as part of the MEMO SML (Frank and Lange, 2007).

feasible to reason about the effects of MetricM on the complexity-reducing effects of using a DSML in performance measurement system design and the complexity-increasing effects of the language itself. Further research is needed to investigate these two opposing effects.

A further limitation of MetricM pertains to its supposedly high costs due to demanding modeling know-how, communication skills, and time requirements. These costs should decline (1) as patterns of certain performance indicator definitions and recombinations of indicator types are identified and conceptualized as reference models, so that model reuse increases and modeling efforts reduce; (2) if indicator modeling is conducted as part of other modeling activities in the context of business process management (Davies et al, 2006), so that existing models, experiences, and know-how can be reused; and (3) if modeling tool support becomes available (for a discussion of these aspects, see Frank et al, 2008, 2009).

7 Conclusion

In part, the present work is motivated by a dilemma. On the one hand, there is clear evidence that controlling complex action systems by focusing on numerical indicators will not only create bias, it may also contribute to opportunistic plans that are not beneficial for an organization's long term competitiveness. On the other hand, there is no doubt that managing complex organizations requires purposeful abstractions, hence some sort of indicators. A method for modeling indicator system is aimed at relaxing this dilemma by rationalizing the process of creating, using and maintaining indicators. This research was conducted following a design research process configured for the epistemological particularity of research on modeling methods to develop a linguistic artifact—a conceptual modeling method—that cultivates the reflective, thoughtful, and rational design, use, and interpretation of performance measurement systems. The method consists of a domain-specific modeling language and corresponding procedural recommendations that guide the application of language concepts. It is built upon and extends an enterprise modeling approach to benefit from the reuse of modeling concepts to provide relevant organizational context including business objectives, organizational roles and responsibilities. Using case studies and prototypical language applications, the study demonstrates how the language supports the design, use, and analysis of performance measurement systems. Results of the method evaluation, which discussed method features in the light of essential domain-specific requirements, concepts and prior work, showed that the presented modeling method meets five essential domain-specific requirements. The method's metamodel-based specification and corresponding graphical notation provides abstract and concrete syntax and formal semantics of dedicated modeling constructs for key domain-specific concepts. The findings indicate that the method allows the convenient creation of coherent and consistent indicator models that promise to facilitate interpretation, judgment, and reflection of performance measurement systems. In particular, the design artifact contributes to existing work in that it adds essential concepts to modeling performance indicators and semantics to key modeling concepts. Implementation of tool support for MetricM is underway in the context of an enterprise modeling environment (Gulden and Frank, 2010). Results of the work on MetricM inform an ongoing multi-annual research project with a leading U.S.-based vendor of IT management software aimed at designing a modeling method for developing advanced management dashboards. In this regard, the artifact outlined in the paper marks a further step toward a comprehensive modeling support for performance measurement. Such a methodical support remains on our research agenda.

References

Abernethy MA, Horne M, Lillis AM, Malina MA, Selto FH (2005) A multi-method approach to building causal performance maps from expert knowledge. Manage Account Res 16(2):135–155

Aichele C (1997) Kennzahlenbasierte Geschäftsprozessanalyse. Gabler, Wiesbaden

Atkinson AA, Waterhouse JH, Wells RB (1997) A stakeholder approach to strategic performance measurement. Sloan Manage Rev 38(3):25–37

Bach V, Brecht L, Hess T, Österle H (1996) Enabling Systematic Business Change : Integrated Methods and Software Tools for Business Process Redesign. Vieweg, Wiesbaden

Bourne M, Mills J, Wilcox M, Neely A, Platts K (2000) Designing, implementing and updating performance measurement systems. Int J Operat Prod Manage 20(7):754–771

Bourne M, Neely A, Mills J, Platts K (2003) Implementing performance measurement systems: a literature review. Int J Bus Perform Manage 5(1):1–24

Bourne M, Kennerley M, Franco-Santos M (2005) Managing through measures: A study of impact on performance. Manage 16(4):373–395

Cardinaels E, van Veen-Dirks PMG (2010) Financial versus non-financial information: The impact of information organization and presentation in a balanced scorecard. Account Organ Soc 35(6):565–578

Chenhall RH, Langfield-Smith K (2007) Multiple perspectives of performance measures. Eur Manage J 25(4):266–282

Davies I, Green P, Rosemann M, Indulska M, Gallo S (2006) How do practitioners use conceptual modeling in practice? Data Knowl Eng 58(3):358–380

De Haas M, Kleingeld A (1999) Multilevel design of performance measurement systems: enhancing strategic dialogue throughout the organization. Manage Account Res 10(3):233–261

Drucker PF (1954) The Practice of Management. Harper & Row, New York

Eccles R, Pyburn P (1992) Creating a comprehensive system to measure performance. Manage Account 74(4):41–44

Eccles RG (1991) The Performance Measurement Manifesto. Harv Bus Rev 69(1):131–137

Edwards JR, Bagozzi RP (2000) On the nature and direction of relationships between constructs and measures. Psychol Methods 5(2):155–174

Epstein M, Manzoni JF (1998) Implementing corporate strategy: from tableaux de bord to balanced scorecards. Eur Manage J 16(2):190–203

Fortuin L (1988) Performance indicators—Why, where and how? Eur J Operat Res 34(1):1–9

Frank U (1994) Multiperspektivische Unternehmensmodellierung: Theoretischer Hintergrund und Entwurf einer objektorientierten Entwicklungsumgebung. Oldenbourg, München

Frank U (1998) Essential Research Strategies in the Information Systems Discipline—Reflections on Formalisation, Contingency and the Social Construction of Reality. The Systemist 20:98–113

Frank U (2002) Multi-Perspective Enterprise Modeling (MEMO): Conceptual framework and modeling languages. In: Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS), Honululu, pp 72–82

Frank U (2005) Contribution to empirical research strategies in conceptual modeling – silver bullet or academic toys? WIRTSCHAFTSINFORMATIK 47(2):153–154

Frank U (2006a) Evaluation of reference models. In: Fettke P, Loos P (eds) Reference Modeling for Business Systems Analysis, Idea Group, Hershey, PA, pp 118–140

Frank U (2006b) Towards a pluralistic conception of research methods in information systems research. ICB Research Report 7, Institute for Computer Science and Business Information Systems (ICB), University of Duisburg-Essen, Germany

Frank U (2008) The MEMO Meta Modelling Language (MML) and Language Architecture. ICB Research Report 24, Institute for Computer Science and Business Information Systems (ICB), University of Duisburg-Essen, Germany

Frank U (2010a) Prolegomena of a Method to Guide the Development of Domain-Specific Modelling Languages, Unpublished Manuscript; to appear in 2011

Frank U (2010b) The MEMO Organisation Modelling Language : An Update, Unpublished Manuscript; to appear in 2011

Frank U, Lange C (2007) E-MEMO: A Method to support the Development of customized Electronic Commerce Systems. Inf Syst E-Bus Manage 5(2):93–116

Frank U, Heise D, Kattenstroth H, Schauer H (2008) Designing and utilising business indicator systems within enterprise models – outline of a method. In: Loos P, Nüttgens M, Turowski K, Werth D (eds) Modellierung betrieblicher Informationssysteme (MobIS 2008), GI, Bonn, Lecture Notes in Informatics, vol 141, pp 89–105

Frank U, Heise D, Kattenstroth H (2009) Use of a domain specific modeling language for realizing versatile dashboards. In: Tolvanen JP, Rossi M, Gray J, Sprinkle J (eds) Proceedings of the 9th OOPSLA workshop on Domain-Specific Modeling (DSM), Helsinki Business School, Helsinki

Gulden J, Frank U (2010) MEMOCenterNG - A Full-featured Modeling Environment for Organization Modeling and Model-driven Software Development. In: Soffer P, Proper E (eds) Proceedings of the CAiSE Forum 2010, Hammamet, Tunisia, June 9–11, 2010, CEUR Workshop Proceedings, vol 592, pp 76–83

Habermas J (1968) Technik und Wissenschaft als Ideologie. In: Habermas J (ed) Technik und Wissenschaft als Ideologie, Frankfurt/M., pp 48–103

Hauser J, Katz G (1998) Metrics: You Are What You Measure! Eur Manage J 16(5):517–528

Henderson-Sellers B, Ralyté J (2010) Situational method engineering: State-of-the-art review. J Univ Comp Sci 16(3):424–478

Hope J (2007) Beyond budgeting to the adaptive organization. In: Neely A (ed) Business Performance Measurement : Unifying theories and integration practice, 2nd edn, Cambridge Univ. Press, pp 163–178

Kaplan RS (2010) Conceptual foundations of the balanced scorecard. Working Paper 10-074, Harvard Business School, Harvard University, Cambridge, Mass., paper originally prepared for C. Chapman, A. Hopwood, and M. Shields (eds.), Handbook of Management Accounting Research: Volume 3 (Elsevier, 2009).

Kaplan RS, Norton DP (1992) The Balanced Scorecard: Measures that Drive Performance. Harv Bus Rev 70:71–79

Kaplan RS, Norton DP (1996a) The Balanced Scorecard : Translating Strategy Into Action. Harvard Business School Press, Boston, Mass.

Kaplan RS, Norton DP (1996b) Linking the balanced scorecard to strategy. Cal Manage Rev 39(1):53–80

Kaplan RS, Norton DP (2004) Strategy Maps : Converting Intangible Assets Into Tangible Outcomes. Harvard Business School Press, Cambridge, Mass.

Kelly S, Tolvanen JP (2008) Domain-Specific Modeling: Enabling Full Code Generation. John Wiley & Sons

Korherr B, List B (2007a) Extending the EPC and the BPMN with business process goals and performance measures. In: 9th International Conference on Enterprise Information Systems, ICEIS 2007, Funchal, Madeira, June 12-16, 2007, Revised Selected Papers, Citeseer, pp 287–294

Korherr B, List B (2007b) Extending the EPC with performance measures. In: Proceedings of the 2007 ACM Symposium on Applied Computing (ACM-SAC'07), ACM, pp 1266–1267

Kronz A (2005) Management von Prozesskennzahlen im Rahmen der ARIS-Methodik. In: Scheer AW, Jost W, Heß H, Kronz A (eds) Corporate Performance Management, Springer, pp 31–44

Lankhorst M (2009) Enterprise Architecture at Work: Modelling, Communication and Analysis, 2nd edn. Springer, Berlin

Lebas M, Euske K (2007) A conceptual and operational delineation of performance. In: Neely A (ed) Business Performance Measurement : Unifying theories and integration practice, 2nd edn, Cambridge Univ. Press, pp 125–139

Lynch RL, Cross KF (1991) Measure up! The Essential Guide to Measuring Business Performance. Mandarin, London

Malina MA, Nørreklit HSO, Selto FH (2007) Relations among measures, climate of control, and performance measurement models. Contemp Account Res 24(3):935–982

Moers F (2005) Discretion and bias in performance evaluation: the impact of diversity and subjectivity. Account Organ Soc 30(1):67–80

Nagel E (1931) Measurement. Erkenntnis 2(1):313–335

Neely A, Gregory M, Platts K (1995) Performance measurement system design : A literature review and research agenda. Int J Operat Prod Manage 15(4):80–116

Neely A, Richards H, Mills J, Platts K, Bourne M (1997) Designing performance measures: a structured approach. Int J Operat Prod Manage 17:1131–1152

Neely A, Mills J, Platts K, Richards H, Gregory M, Bourne M, Kennerley M (2000) Performance measurement system design: developing and testing a process-based approach. Int J Operat Prod Manage 20(9/10):1119–1145

Neely A, Kennerley M, Adams C (2007) Performance measurement frameworks: a review. In: Neely A (ed) Business Performance Measurement : Unifying theories and integration practice, 2nd edn, Cambridge Univ. Press, pp 143–162

Neely AD (2007) Business performance measurement, 2nd edn. Cambridge Univ. Press, Cambridge, U.K.

Nørreklit H (2000) The balance on the balanced scorecard: A critical analysis of some of its assumptions. Manage Account Res 11:65–88

Nørreklit H (2003) The Balanced Scorecard: What is the score? A rhetorical analysis of the balanced scorecard. Account Organ Soc 28(6):591–619

Nørreklit H, Nørreklit L, Falconer M (2007) Theoretical conditions for validity in accounting performance measurement. In: Neely A (ed) Business Performance Measurement : Unifying theories and integration practice, 2nd edn, Cambridge Univ. Press, pp 179–217

Nørreklit L, Nørreklit H, Israelsen P (2006) The validity of management control topoi: Towards constructivist pragmatism. Manage Account Res 17(1):42–71

Ortner E (2008) Language-critical enterprise and software engineering. In: Proceedings of the Fourteenth Americas Conference of Information Systems, AMCIS 2008, Toronto, ON, Canada, Aug 14–17, 2008

Österle H (1995) Business Engineering. Prozeß- und Systementwicklung. Springer

Österle H, Brenner C, Gaßner C, Gutzwiller T, Hess T (1996) Business Engineering – Prozeß- und Systementwicklung. Band 2: Fallbeispiel. Springer, Berlin

Österle H, Becker J, Frank U, Hess T, Karagiannis D, Krcmar H, Loos P, Mertens P, Oberweis A, Sinz EJ (2010) Memorandum on design-oriented information systems research. Eur J Inf Syst 20:7–10

Palpanas T, Chowdhary P, Mihaila G, Pinel F (2007) Integrated model-driven dashboard development. Inf Syst Front 9(2–3):195–208

Peffers K, Tuunanen T, Rothenberger MA, Chatterjee S (2007) A design science research methodology for information systems research. Journal of Management Information Systems 24(3):45–77

Perrin B (1998) Effective Use and Misuse of Performance Measurement. Am J Eval 19(3):367–379

Pfeffer J (1981) Management as symbolic action. In: Cummings LL, Staw BM (eds) Research in Organizational Behavior, vol 3, JAI Press, Greenwich, Conn., pp 1–52

Pike S, Roos G (2007) The validity of measurement frameworks: measurement theory. In: Neely A (ed) Business Performance Measurement : Unifying theories and integration practice, 2nd edn, Cambridge Univ. Press, pp 218–235

Popova V, Sharpanskykh A (2010) Modeling organizational performance indicators. Inf Syst 35(4):505–527

Pourshahid A, Chen P, Amyot D, Weiss M, Forster A (2007) Business Process Monitoring and Alignment: An Approach Based on the User Requirements Notation and Business Intelligence Tools. In: 10th Workshop of Requirement Engineering (WERE'07), Toronto, pp 80–91

Pourshahid A, Amyot D, Peyton L, Ghanavati S, Chen P, Weiss M, Forster AJ (2008) Toward an Integrated User Requirements Notation Framework and Tool for Business Process Management. In: 2008 International MCETECH Conference on e-Technologies, pp 3–15

Ridgway VF (1956) Dysfunctional consequences of performance measurements. Adm Sci Quart 1(2):240–247

Rolland C (2007) Method engineering: Trends and challenges. In: Ralyté J, Brinkkemper S, Henderson-Sellers B (eds) Situational Method Engineering, Springer, IFIP, vol 244, p 6

Ronaghi F (2005) A modeling method for integrated performance management. In: Proceedings of the Sixteenth International Workshop on Database and Expert Systems Applications (DEXA'05), pp 972–976

Rosanas JM (2008) Beyond economic criteria: A humanistic approach to organizational survival. J Bus Ethics 78(3):447–462

Rosanas JM, Velilla M (2005) The Ethics of Management Control Systems: Developing Technical and Moral Values. J Bus Ethics 57(1):83–96

Rossi M, Ramesh B, Lyytinen K, Tolvanen JP (2004) Managing evolutionary method engineering by method rationale. J Assoc Inform Syst 5(9):356–391

Scheer AW (1992) Architecture of Integrated Information Systems : Foundations of Enterprise Modelling. Springer, Berlin

Scheer AW (2000) ARIS : Business Process Modeling, 3rd edn. Springer, Berlin, Heidelberg

Simon HA (1964) On the concept of organizational goal. Adm Sci Quart 9(1):1–22

Simons R (1995) Levers of Control: How Managers Use Innovative Control Systems to Drive Strategic Renewal. Harvard Business School Press, Boston, Mass.

Simons R, Davila A, Kaplan RS (2000) Performance Measurement & Control Systems for Implementing Strategy. Prentice Hall

Smith Stevens S (1959) Measurement. In: Churchman CW (ed) Measurement: Definitions and Theories, John Wiley & Sons, New York, pp 18–36

Speckbacher G, Bischof J, Pfeiffer T (2003) A descriptive analysis on the implementation of balanced scorecards in german-speaking countries. Manage Account Res 14(4):361–388

Strembeck M, Zdun U (2009) An approach for the systematic development of domain-specific languages. Softw Pract Exper 39(15):1253–1292

Taticchi P (2010) Business Performance Measurement and Management: New Contexts, Themes and Challenges. Springer, Berlin

Townley B, Cooper DJ, Oakes L (2003) Performance measures and the rationalization of organizations. Organ Stud 24(7):1045

Tuomela TS (2005) The interplay of different levers of control: A case study of introducing a new performance measurement system. Manage Account Res 16(3):293–320

Verschuren P, Hartog R (2005) Evaluation in design-oriented research. Qual Quant 39(6):733–762

von Wright GH (1971) Explanation and Understanding. Cornell Univ Pr

Wand Y, Weber R (2002) Research commentary: information systems and conceptual modeling — a research agenda. Inf Syst Res 13(4):363–376

Wand Y, Monarchi DE, Parsons J, Woo CC (1995) Theoretical foundations for conceptual modelling in information systems development. Dec Supp Syst 15(4):285–304

Weick KE (1979) Cognitive processes in organizations. In: Staw BM (ed) Research in Organizational Behavior, vol 1, JAI Pres, Greenwich, Conn., pp 41–74

Weick KE (1980) The Social Psychology of Organizing, 2nd edn. McGraw-Hill Higher Education, New York