TECHNISCHE UNIVERSITÄT MÜNCHEN

Lehrstuhl für Informatik XIX

Integrated Software Support for Quantitative Models in the Domain of Enterprise Architecture Management

Ivan B. Monahov

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender:		Univ.-Prof. Dr. J. Schlichter
Prüfer der Dissertation:	1.	Univ.-Prof. Dr. F. Matthes
	2.	Univ.-Prof. Dr. A. Pretschner

Die Dissertation wurde am 02.06.2014 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 13.10.2014 angenommen.

Zusammenfassung

Wie in der Literatur beschrieben, setzen bestehende Rahmenwerke hauptsächlich qualitative Modelle ein, um die zugrundeliegende Unternehmensarchitektur (UA) und ihre Managementfunktion (UAM) zu bewerten und zu analysieren. Durch den wachsenden Reifegrad und den steigenden finanziellen Druck auf UA Initiativen, benötigen Unternehmensarchitekten zusätzlich quantitative Modelle (Metriken), um Ihren Interessenvertretern relevante Informationen in aggregierter Form zur Verfügung zu stellen. Zusätzlich sind Unternehmensarchitekten daran interessiert Leistungsmessungen für das UAM zu etablieren und nachzuverfolgen. Metriken ermöglichen außerdem Messungen hinsichtlich der Erreichung vordefinierter UAM Ziele und unterstützen bei der Modellierung und Analyse komplexer und unerwarteter Entwicklungen. In dieser Arbeit zeigen wir, wie Metriken die Bewertung statischer UA Aspekte, w. z. B. Heterogenität und Standardkonformität, unterstützen können. Wir beschreiben wie Metriken zur Steuerung und Planung von Transformationsprojekten verwendet werden können und diskutieren ihre Verwendung für UAM Leistungsbewertungen, z. B. Reaktionszeiten und Durchsatz.

Im konzeptionellen Teil der Arbeit nehmen wir eine systemische Perspektive auf ein Unternehmen ein und präsentieren eine einheitliche Terminologie sowie vier Nutzungsszenarien für Metriken unter Beachtung existierender Arbeiten in der UAM Domäne. Wir entwickeln eine Liste von Risiken und empfohlenen Gegenmaßnahmen, verbunden mit dem Einsatz von Metriken, und schlagen eine anpassbare Struktur für die Dokumentation, Beschreibung und das Auffinden von Metriken vor. Die Struktur besteht aus zehn Elementen, welche in zwei Kategorien aufgeteilt sind – allgemeine und organisationsspezifische Beschreibungselemente. Unter Verwendung dieser Struktur präsentieren wir einen Metrikenkatalog, basierend auf Beobachtungen in deutschen Unternehmen. Weiterhin stellen wir eine Methode vor, die ein ganzheitliches Lebenszyklusmanagement organisationsspezifischer Metriken ermöglicht. Die Methode lässt sich außerdem in Muster-basierte und agile UAM Rahmenwerke, w. z. B. BEAMS, integrieren. Diese Integration ermöglicht es Unternehmen systematisch und unter Berücksichtigung des jeweiligen Unternehmenskontexts und der Ziele der beteiligten Interessensvertreter geeignete UAM Metriken zu definieren und zu implementieren. Zusätzlich ermöglicht diese Integration das Vermitteln neuen Wissens durch die einheitliche Dokumentation neuer Metriken in der BEAMS Wissensbasis. Durch das Beobachten der Verwendung und der Evolution der Wissensbasis können mit der Zeit praxiserprobte Metriken sowie Metrik-Antimuster identifiziert werden.

Im Implementierungsteil der Arbeit präsentieren wir einen Softwareprototyp, der die vorgestellten Konzepte unterstützt. Zu diesem Zweck definieren wir eine domänenspezifische Sprache mit minimaler Anzahl an Operatoren und Konstrukten, die für die Umsetzung der Metriken aus dem Katalog benötigt werden. Weiterhin präsentieren wir eine Implementierung und Integration dieser Sprache innerhalb eines kommerziellen UAM Werkzeugs. Wir diskutieren ausgewählte Implementierungsaspekte, w. z. B. die Implementierung eines Typsystems zur Unterstützung des Beziehungsmanagements von Metriken und der Reaktion auf Änderungen im zugrundeliegenden UA Model zur Laufzeit. Abschließend zeigen wir den Mehrwert der Lösung für Unternehmensarchitekten basierend auf den Ergebnissen einer Fallstudie bei einem deutschen Finanzinstitut. Dabei fassen wir die qualitativen und subjektiven Bewertungen der beteiligten Experten zusammen.

Abstract

As described by literature, the existing Enterprise Architecture (EA) management frameworks employ mainly qualitative models to assess and analyze the underlying EA and its management function. However, with growing maturity and increasing financial pressure on EA initiatives, enterprise architects require in addition quantitative models (metrics) to provide relevant and aggregated information to their stakeholders. Further, enterprise architects are interested in defining and monitoring EA management performance itself. Finally, metrics support the measurement of the achievement of predefined EA management goals and the modeling and analysis of complex and unexpected developments. In this thesis, we show how metrics can support the assessment of static aspects of the EA, e.g., heterogeneity and standard conformity. We outline how metrics can be used for the steering and planning of transformation projects and we discuss how metrics can be used for the management of performance measurements, e.g., reaction times and throughput.

In the conceptual part of the thesis, by taking a systemic view on an enterprise, we firstly propose a terminology base for the usage of metrics in the domain of EA management. We present four distinct usage scenarios for metrics under consideration of the existing work in the domain. In addition, we develop a comprehensive list of risks and recommended countermeasures associated with the usage of metrics in this domain. We propose a uniform and configurable structure for the documentation, description and retrieval of metrics comprising ten elements organized in two categories - general and organization-specific metric description elements. Using this structure, we present a catalog of metrics, which we observed in German industry. Furthermore, we introduce a life-cycle management method for organization-specific metrics using our documentation structure and catalog. This method also fits pattern-based and agile EA management frameworks like the BEAMS framework. Thereby, the method enables organizations to systemically identify and define metrics tailored to the specific needs of their stakeholders and their organizational context. Additionally, the documentation of new metrics in the BEAMS method base supports the communication of new knowledge to other interested organizations. By monitoring the usage and the evolution of the method base, metrics best-practices as well as metric anti-pattern for specific EA management problems can be identified over time.

In the implementation part of the thesis, we present a software prototype supporting the previously introduced concepts. For this purpose we have designed and implemented a domain-specific language (DSL) with a minimal number of operators and constructs required to implement the metric best-practices from the method base. Further, we present a concrete implementation of this DSL within a given commercial EA management tool. We discuss the most challenging and interesting aspects of the implementation from a software engineering perspective, e.g. the implementation of a type checker to support the management of metric relationships and to account for changes at instance and schema level of the underlying EA model at runtime. Finally, to demonstrate the benefits of our solution for enterprise architects, we describe a concrete application case study of the tool in the context of a German financial organization and summarize the user feedback gathered during the introduction and implementation phase to guide future research.

Acknowledgment

This thesis summarizes the main results of my research carrier at the Chair for Software Engineering for Business Information Systems (sebis) at the Technische Universität München. Consequently, many of the presented ideas, solutions and concepts were born or benefited from discussions and collaborations with many people, who I want to thank.

I want to express my sincerest gratitude to my primary supervisor, Prof. Dr. Florian Matthes, for giving me the opportunity to conduct research in the fascinating domain of EA management, as well as for his great support, valuable feedback, and challenging advices while supervising my work. Further I want to express my sincere gratitude to Prof. Dr. Alexander Pretschner for his constructive and valuable feedback as a second supervisor of my thesis.

One of the main success factors for the successful completion of this thesis was the excellent collaboration with my colleagues at the chair. Hence, I want to thank all of them for their patience in several research discussion rounds, their support in writing publications, reviewing documents and discussing possible improvements. These collaborations deeply influenced and improved my work. In particular, I want to express my gratitude to Dr. Christian M. Schweda, Dr. Sabine Buckl, Dr. Christopher Schulz, Alexander W. Schneider, Thomas Reschenhofer, Matheus Hauder, Klym Schumaiev, and Sascha Roth. In addition, I want to thank Dr. Marco Kuhrmann and Dr. Daniel Méndez Fernández for sharing their experience regarding different software engineering and research methodology aspects.

My work would not be successful without the valuable contributions of our students. Hence, I want to thank Thomas Reschenhofer for his contributions in the development of MxL and TxL, Erdisa Subashi for her contribution in the research concerned with risks and countermeasures, Michael Schätzlein for his contribution for integrated metric visualizations in our prototype, Manoj Mahabaleshwar for his contribution regarding the EA model history access in MxL expressions, and Jawid Rassa for his support during the evaluation of our holistic metric life-cycle management method in a real-life case study.

Discussing my ideas, results and opinions with practitioners and applying the prototype in parallel research projects and in a real world case study strongly influenced my work. Hence, I want to thank all industry experts who took their time to discuss my work, to share their experience with me, and to help me to apply my prototype in practice. In particular, I want to express my gratitude to Mr. Gerhard Klausen, who shared his experience and thoughts in many discussion rounds and provided valuable data and feedback for my research.

Finally, I express my sincere gratitude to my partner Regine Preuß for her patience and understanding for the long days and nights I spend in front of my computer writing this thesis, the valuable discussions regarding my results, the reviewing of this document and foremost for her deep love and for keeping on motivating me all the time.

Garching b. München, November 20, 2014

Ivan Monahov

Contents

1 Motivation 1
1.1 Research Questions 5
1.2 Research Method 7
1.3 Structure of the Thesis 9

2 Introduction and Related Work 11
2.1 Introduction 11
2.1.1 Enterprise Architecture 11
2.1.2 Enterprise Architecture Management 13
2.1.3 Building Blocks for Enterprise Architecture Management Solutions . . . 14
2.2 Fundamental Concepts for Metrics in Management 16
2.3 Related Work 21
2.3.1 Metrics for Application Landscapes by Lankes 25
2.3.2 Assessment of Applications in the Context of Their EA by Addicks . . . 27
2.3.3 Value-oriented EA Management by Stutz 28
2.3.4 The Enterprise Architecture Value Framework by Plessius et al. 31
2.3.5 Predictive, Probabilistic Architecture Modeling Framework by Johnson et al. 32
2.3.6 Enterprise Architecture Benefits by Niemi 33
2.3.7 A DSL-based Method for Performance Measurement by Strecker et al. . 35
2.3.8 Summary 36

3 Concept Development 39
3.1 Identified Usage Scenarios 39
3.2 Terminology 45
3.3 Risks and Countermeasures 52
3.4 Metric Management Fact Sheet 65
3.4.1 Metric Fact Sheets in IT Management Literature 66
3.4.2 Metric Fact Sheets in Enterprise Management Literature 67
3.4.3 Metric Fact Sheets in EA Management Literature 72

3.4.4 A Generic Metric Management Fact Sheet (MMFS) 76
3.4.5 MMFS Evaluation 86
3.5 Metric Management Catalog 90
3.6 Metric Management Method 95

4 Metric Management Method Evaluation 103
4.1 Evaluation 103
4.1.1 Method Evaluation via Expert Interviews 103
4.1.2 Application of the Method in Industry 105
4.2 Critical Reflection 110

5 Integrated Software Support 115
5.1 Tricia 115
5.1.1 Models in Tricia 116
5.1.2 Controllers and Views in Tricia 117
5.1.3 Hybrid Wikis 118
5.1.4 Outline of the Desired Integrated Software Solution 120
5.2 MxL 1.0 Design and its TxL 1.0 Implementation 122
5.2.1 Evaluation of the TxL 1.0 Prototype 125
5.2.2 Shortcomings of TxL 1.0 129
5.3 MxL 2.0 Design and its TxL 2.0 Implementation 130
5.3.1 Types in MxL 2.0 130
5.3.2 Operators in MxL 2.0 132
5.3.3 MxL 2.0 Interpreter 137
5.4 Selected Implementation Aspects of TxL 2.0 141
5.4.1 Derived Attributes and Custom Functions 141
5.4.2 Compile-time Analysis of Expressions 142
5.4.3 Access to Model History 143
5.4.4 User-defined Metric Visualizations 145
5.4.5 Web-based Editor Support for Expressions 153
5.4.6 Implementation of the MMFS in Tricia 155

6 Evaluation 161
6.1 Structure of the Case Study 161
6.2 Expert Interview Design and Results 162
6.2.1 MxL 2.0 Design Evaluation 163
6.2.2 Evaluation of TxL 2.0 and the Related Tricia Extensions 165
6.2.3 Evaluation of the Provided Software Support for our Holistic Metric Life-cycle Management Method 167
6.2.4 Summary 170
6.2.5 Threats to Validity 171

7 Critical Reflection and Future Research 173
7.1 Conclusion 173
7.2 Critical Reflection and Future Research 174
7.2.1 Authorization in MxL Functions 174
7.2.2 Identity Evaluation in MxL 175

7.2.3 Evaluation Strategy of MxL Functions 176
7.2.4 Version Control of MxL Functions 178
7.2.5 Import of EA Model History Data 178
7.2.6 Integration of the MMFS Structure as a Tricia Object 179
7.2.7 Investigating MxL from Computability Theory Perspective 181

A Questions List for the Evaluation **183**

Bibliography **187**

List of Figures

1.1 Outline of the thesis linked to the sub-contributions, cf. Table 1.1 9

2.1 Overall structure of the enterprise architectureaccording to [Bu11, Sc11] 12
2.2 BEAMS activity framework according to Buckl [Bu11] and Schweda [Sc11] . . . 14
2.3 The BEAMS framework according to [Bu11, Sc11] 16
2.4 Types of metrics according to Siegwart et al. [SRS10] 18
2.5 ROE formula according to [SRS10] 19
2.6 The basic controlling loop according to Kütz [Kü10] 20
2.7 Steering object and it's controlling value vectors according to Kütz [Kü10] . . . 21
2.8 An effective literature review process according to Levy et al. [LE06] 21
2.9 Exemplary visualization of failure propagation metrics according to Lankes [La08] 26
2.10 The ITEVA method for assessment of applications in the context of their EA according to Addicks [Ad10] 27
2.11 Value oriented EA management method according to Stutz [St09b] 28
2.12 Ideal value based EA management process according to Stutz [St09b] 30
2.13 Overview of the EAVF according to Plessius et al. [PSP12] 31
2.14 P^2AMF method according to Buschle et al [BJS13] 32
2.15 EA benefits categorization according to Niemi [Ni08] 34
2.16 The MetricML meta-model according to Strecker et al. [St12, p. 18] 36
2.17 Incorporated concepts from related works in our solution 37

3.1 Casual loop diagram for the adoption rate of a new product based on Sterman [St01] 40
3.2 Stock and flow diagram for the adoption rate of a new product based on [St01] 41
3.3 The system dynamics process according to Forrester [Fo94] 41
3.4 A systemic view on an enterprise [MRM13] 43
3.5 Usage scenarios for quantitative models (metrics) 44
3.6 Conceptualization of an EA metric 48
3.7 Minimal EA model for the example EA metric [Ma12a] 49
3.8 Conceptualization of a performance indicator 50

3.9 Overview of the thesis's terminology 51
3.10 The hermeneutic cycle according to Heidegger [He96] 53
3.11 Overview of typical drivers for the usage of EA management metrics 58
3.12 Stakeholders interested in the usage of EA management metrics 59
3.13 Overview of typical benefits associated with the usage of EA management metrics 61
3.14 Results of the risks prioritization 63
3.15 IT controlling metric description structure according to Kütz [Kü10] 66
3.16 Management metric description structure according to Parmenter [Pa10] 68
3.17 Management metric description structure according to Popova et al. [Pa10] . . 69
3.18 Management metric description structure according to Neely et al. [NAK02] . . 71
3.19 EA metric description structure according to Addicks [Ad10] 72
3.20 BSC-based description structure for EA management metrics according to Stutz [St09b] 74
3.21 Metric description structure according to Stecker et al. [St12] 75
3.22 Overview of the MMFS and its elements 77
3.23 MMFS application for the documentation of an EA metric 80
3.24 Excerpt of the Goal-Concern-List according to our catalog [Ma12a] 92
3.25 Integrated information model based on the 52 catalog metrics mapped to EA layers and cross-cutting functions according to [Ma12a], cf. Figure 2.1 93
3.26 Holistic life-cycle metric management method 96

4.1 Example of an adapted metric during the application of our method according to [Ra13] 108
4.2 Integrated information model of the metrics defined in the case study according to [Ra13] 109

5.1 UML class diagram of Tricia's basic model hierarchy according to [Re13] 116
5.2 A Tricia page and its basic parts according to [Re13] 117
5.3 HTTP request processing by Tricia according to [Re13] 118
5.4 Example of a type definition *Employee* consisting of four property definitions according to [Re13] 118
5.5 UML class diagram of the abstract Hybrid wikis data model according Matthes et al. [MNS11]. 119
5.6 Implementation of the MMFS structure 120
5.7 Hybrid implementation of metrics information models 120
5.8 Implementation of the metric catalog as a Tricia workspace 121
5.9 Implementation of the calculation rule of the catalog's metric *Application continuity plan availability* [Ma12a, p. 19] as MxL 1.0 custom function 124
5.10 An excerpt of the SmartNet information model according to [MRM13] 126
5.11 An excerpt of the SmartNet Navigator of an exemplary project according to [MRM13] 127
5.12 TxL 1.0 implementation of a process phase's status according to [MRM13] . . . 127
5.13 Execution of the queries and visualization functions of the SmartNet Navigator according to [MRM13] 128
5.14 Type hierarchy of MxL 2.0 for its integration in Tricia according to [Re13] . . . 132
5.15 The hierarchy of MxL's 2.0 elements according to [Re13] 137
5.16 Overview of MxL's 2.0 interpreting and evaluating process according to [Re13] . 138

5.17 Overview of the type checking process (*FunctionParameterTypeChecker* according to [Re13]) . . . 139
5.18 The MxL connector and subcomponents according to [Re13] . . . 139
5.19 Representation of derived attributes in Tricia according to [Re13] . . . 141
5.20 Tricia's view for the management of derived attributes and custom functions according to [Re13] . . . 142
5.21 Example of a TxL 2.0 query . . . 143
5.22 Example of a TxL 2.0 query using the @ operator . . . 144
5.23 Example for historicization change sets of a Tricia model according to [Ma14] . 144
5.24 Example of a line chart visualization according to [Sc14] . . . 146
5.25 Example of an area chart visualization according to [Sc14] . . . 146
5.26 Example of a column chart visualization according to [Sc14] . . . 147
5.27 Example of a bar chart visualization according to [Sc14] . . . 147
5.28 Example of a column chart for the visualization of two metrics according to [Sc14]148
5.29 Example of a pie chart visualization according to [Sc14] . . . 148
5.30 Usage of a column chart instead of a pie chart (cf. Figure 5.29) according to [Sc14]149
5.31 Example of a kiviat chart visualization according to [Sc14] . . . 149
5.32 Usage of a bar chart instead of a kiviat chart (cf. Figure 5.31) according to [Sc14]150
5.33 Example of a gauge visualization for the metric *Application continuity plan availability* from the catalog according to [Sc14] . . . 150
5.34 Example of a colored bullet chart visualization according to [Sc14] . . . 150
5.35 Extension of Tricia's rich text editor by a button allowing user-defined metric visualizations . . . 151
5.36 Selecting metric visualization types in Tricia at runtime according to [Sc14] . . 151
5.37 Entering the name and target value of a custom function to define a corresponding visualization at runtime according to [Sc14] . . . 152
5.38 Adjusting specific visualization properties for a selected metric visualization at runtime according to [Sc14] . . . 152
5.39 Mapping of configuration properties to a line chart visualization according to [Sc14] . . . 153
5.40 Calling the MxL code editor from Tricia's rich-text editor according to [Re13] . 154
5.41 Writing a TxL 2.0 expression at runtime according to [RMM14] . . . 154
5.42 Extension of Tricia's *TypeDefinition* by the properties *PageTemplate* and *PageTemplateCSS* according to [Re13] . . . 155
5.43 The *Page Template* view of the exemplary type definition *Employee*, whereas neither a template nor CSS classes are defined according to [RMM14] . . . 156
5.44 *Add/Edit property* button for defining page templates according to [RMM14] . . 156
5.45 Template definition for the type *metric description* according to [RMM14] . . . 157
5.46 Representation of the metric *Application continuity plan availability* based on the layout definition for the type *metric description* according to [RMM14] . . . 158
5.47 MxL implementation from the Tricia version of our catalog . . . 159

6.1 Structure of the case study . . . 162
6.2 MMFS description of the observed new EA management metric in the case study169

List of Tables

1.1 Overview of the sub-contributions of this thesis 4

2.1 Example of base rates usage (based on Siegwart et al. [SRS10]) 18
2.2 Output overview from EA management literature 24
2.3 Output overview from IT management literature 24
2.4 Output overview from enterprise management literature 25

3.1 Overview of the most frequently used terms in the related sources 46
3.2 Overview of general risks and related countermeasures 54
3.3 Overview of data related risks and suggested countermeasures 54
3.4 Overview of risks and countermeasures related to organizational goals 55
3.5 Overview of risks and countermeasures related to target values of goals 55
3.6 Overview of risks and countermeasures related to the number of used metrics . 55
3.7 Overview of risks and countermeasures related to ethical aspects 55
3.8 Overview of risks and countermeasures related to rewards associated with metrics 56
3.9 Overview of risks and countermeasures related to the presentation of metrics . . 56
3.10 Overview of typical stakeholders and a mapping to more general stakeholder terms 59
3.11 Typical benefits associated with the usage of EA management metrics as mentioned by the participants and a mapping to more general benefits 60
3.12 EA management metric examples provided by the participants and a corresponding mapping to typical EA elements 62
3.14 Mapping of MMFS elements to elements suggested by literature 81
3.16 Participants metric design experience and metric usage 88
3.18 Results from the appropriateness evaluation of the MMFS elements 89
3.20 Excerpt of the Goal-Metric-Matrix according to our catalog [Ma12a] 92

4.1 BSC perspectives, goals, and mapping to BEAMS goals 106

5.1 Color encoding and status aggregations of a project, a process phase, a module, and an activity type . . . 126
5.2 Basic types in MxL 2.0 . . . 131
5.3 Arithmetic operators in MxL 2.0 . . . 132
5.4 Comparison and logical operators in MxL 2.0 . . . 133
5.5 Common query functions in MxL 2.0. . . . 134
5.6 Quantifier functions in MxL 2.0 . . . 134
5.7 Set functions in MxL 2.0 . . . 135
5.8 Element functions in MxL 2.0. . . . 135
5.9 Partitioning functions in MxL 2.0. . . . 135
5.10 Aggregation functions in MxL 2.0. . . . 136

A.1 Complete list of questions asked in the thesis' evaluation . . . 185

CHAPTER 1

Motivation

Modern organizations are forced to respond to a rapidly growing number of changes in their business and technological environment. Typical drivers for these changes are the high dynamics in today's world markets, the increasing speed of technological progress, and the expanding number of restrictions defined by different legal and regulatory bodies. Consequently, both—the business and the IT architecture of an organization are affected by these changes in different ways and need to be aligned to ensure proper, effective and efficient IT support for the business.

As a commonly accepted technique to accomplish this task, organizations employ *Enterprise Architecture* (EA) management. This discipline seeks to create a holistic perspective on an enterprise comprising both—business and IT elements. Further, EA management fosters the communication by defining a common language for multidisciplinary stakeholders. In addition, EA management links information from differing enterprise data sources to provide a consistent decision making base for the involved stakeholders. Thereby, many well-understood qualitative models (e.g. domain models and process support maps) are employed and maintained by enterprise architects in collaboration with their stakeholders to support and guide the intended organizational change.

As literature confirms (cf. Buckl et al. [BS11]), the existing EA management frameworks employ mainly qualitative models to assess the underlying EA and its management function. However, with growing maturity and increasing financial pressure on EA initiatives, enterprise architects require quantitative models (metrics) to quickly provide relevant and highly aggregated information. This information is required as base for decision making on the one hand and as foundation for EA management performance measurement on the other hand (cf. Kaisler et al. [KAV05], Lucke et al. [LKL10]). As our literature review regarding the current usage of quantitative models shows (cf. Section 2.3), these are applied to support specific EA management usage scenarios. Thereby, quantitative models are employed to analyze and understand unexpected changes in a given EA (cf. Lankes [La08], Addicks [Ad10]). Further, by the application of balanced scorecard approaches, quantitative models are used to support EA management process controlling (cf. Stutz [St09b], Plessius et al. [PSP12]). Moreover, by the application of simulation techniques and uncertainty consideration, quantitative models are used to support both—scenario building, concerned with the improvement of predefined EA properties, and the quantification of predefined behavior aspects of the EA (cf. Johnson et al. [Jo13a], Buschle et al. [BJS13]).

Problem Statement

As described in literature and confirmed by industry experts, enterprise architects require a software supported method to establish a holistic EA metric life-cycle management, tailored to the needs of their stakeholders and aligned with the existing EA management function in their organizations. Thereby, the term holistic covers the process from the systemic definition of metrics under consideration of existing metric best-practices for specific EA management problems, over their organization-specific configuration and instantiation to their implementation and deletion within an EA management tool.

Solution

In this thesis, we present a holistic EA metric life-cycle management method, integrated within the BEAMS framework. This integration allows enterprise architects to manage both—organization-specific qualitative and quantitative EA models under consideration of best-practices in our domain and aligned with the demands of the involved stakeholders. For the design of the method, we consolidate known usage scenarios for metrics in the domain by taking a systemic perspective on an enterprise. Further, we propose a terminology base and a comprehensive list of risks and recommended countermeasures related to the usage of EA management metrics. Furthermore, we present a (minimalistic) metric management fact sheet (MMFS), allowing a comprehensive documentation of metrics in our domain, and supporting their organization-specific configuration and instantiation. Moreover, using this MMFS structure, we present an organized collection (catalog) of metric best-practices as observed in industry. This catalog supports enterprise architects to quickly retrieve recommended metrics by given EA management goals and concerns.

To address the given problem, we further introduce a prototypical software support for our method. For this purpose, we present a new minimalistic, domain-specific, and model-based query language, providing a sufficient number of language constructs to implement the catalog's metrics. Further, we integrate this language in the commercial EA management tool Tricia, which we in turn extended by specific features (e.g. user-defined metric visualizations and web-based user support for the usage of the language) to ensure a sufficient implementation of our method. To demonstrate the usability and benefits provided by our solution, we present the results from a prototypical application of the software, as well as the subjective and qualitative feedback of two enterprise architects from a German financial institution involved in a corresponding case study. Nevertheless, our prototype currently faces specific shortcomings, e.g. limited user access-rights and version control management capabilities, which we consequently describe in detail and which have to be taken into account, if a productive usage of the prototype is intended.

Contribution of the Thesis

The main contribution of this thesis is *a software supported method, allowing the definition of a holistic metric life-cycle management for organizations with an established EA management function*. The method enables enterprise architects as well as other stakeholders to define and manage organization-specific metrics according to their goals and concerns. By the integration of our solution within a given commercial EA management tool, organizations are enabled to manage both—their qualitative and quantitative EA management models within one tool. During the research process to enable this overall contribution, seven minor contributions were made, and several researchers and students were involved as described in Table 1.1.

ID	Name	Brief description
C1	Overview of usage scenarios for metrics in the domain	Based on the consideration of related literature sources from the EA management domain, this thesis provides a consolidated overview of four metric usage scenarios from a systemic viewpoint on an enterprise (cf. Section 3.1).
C2	Proposition of a comprehensive terminology base for the domain	Since a consistent terminology base for our domain does currently not exist in literature and the number of publications concerned with the application of quantitative models is increasing, this thesis proposes a comprehensive terminology for the usage of quantitative models in our domain (cf. Section 3.2). In the course of this research, Erdisa Subashi provided valuable contributions by her master's thesis.
C3	Overview of related risks and recommended countermeasures	Accounting for risks and recommended countermeasures in related management fields, this thesis provides a prioritized list of 26 risks and related countermeasures for the domain of EA management based on the results of an expert survey with 14 enterprise architects from German industry (cf. Section 3.3). In the course of this research, Erdisa Subashi provided valuable contributions by her master's thesis, in particular with the identification, consolidation, and evaluation of the survey's results.
C4	Minimal and comprehensive metric description structure for the domain	Although metric management fact sheets (MMFSs) are accepted in related management disciplines, in our domain, a corresponding structure is not available. This thesis introduces a generic MMFS allowing a comprehensive metric description on the one hand, and containing a minimal number of description elements on the other hand. Further, the proposed structure supports the organization-specific configuration and instantiation of metrics in the context of a given user organization (cf. Section 3.4). In the course of this research project, in particular our colleagues Alexander W. Schneider and Dr. Christopher Schulz provided valuable contributions.

ID	Name	Brief description
C5	Organized collection of metric best-practices observed in industry for the domain	Hence a collection of metric best-practices for our domain does currently not exist in literature, in this thesis, we present an organized collection of metrics observed in German industry and structured by our MMFS. This artifact allows the timely retrieval of recommended metric best-practices based on related EA management goals and concerns (cf. Section 3.5). In the course of this project, in particular our colleagues Alexander W. Schneider and Dr. Christopher Schulz provided valuable contributions.
C6	Holistic metric life-cycle management method for the domain	This thesis introduces a holistic metric life-cycle management method and presents an integration of the method into the BEAMS framework to empower the development of organization-specific quantitative EA models (cf. Section 3.6). The base for this method - the BEAMS framework, was provided by Dr. Christian M. Schweda and Dr. Sabine Buckl. For the design of the method, in particular Alexander W. Schneider and Dr. Christopher Schulz provided valuable contributions.
C7	A minimalistic domain-specific model-based query language (MxL) for the implementation of metrics in the domain	Following the recommendations from related literature, this thesis proposes a domain-specific model-based query language (MxL) designed to support the implementation of the metric best-practices from the catalog, comprising a minimal number of basic functions and types required to accomplish this task. By its design, the proposed language can be integrated in an arbitrary EA management tool by implementing a specific integration interface (cf. Section 5.2 and Section 5.3). For this research project, in particular our students Thomas Reschenhofer, Michael Schätzlein and Manoj Mahabaleshwar provided valuable contributions by their master's theses.

Table 1.1.: Overview of the sub-contributions of this thesis

With respect to the existing literature, the following research sub-artifacts are considered as innovative contributions:

1. In existing literature, a heterogeneous terminology for quantitative models is used by different author groups. In addition different perspectives on an enterprise are taken for the usage of metrics. Hence, based on the idea of taking a systemic perspective on an enterprise and accounting for different terms as well as known usage scenarios for metrics, we present a corresponding terminology base for our domain.

2. A comprehensive list of risks and countermeasures related to the usage of quantitative models in our domain is currently not described in literature. Hence, accounting for relevant knowledge in related management fields (e.g. IT controlling and enterprise management), we propose a comprehensive list of risks and countermeasures for our domain.

The appropriateness, relevance, and completeness of these risks and countermeasures are confirmed by enterprise architects form industry.

3. In current EA management literature, a generic and minimal description structure for EA management metrics is not available. Hence, under consideration of description structures from related management fields, we present a metric description structure tailored to its usage in our domain. Further, this structure consists of the minimal number of description elements, required to ensure a comprehensive metric description in our understanding. Furthermore, the structure supports the organization-specific configuration and instantiation of metrics. Currently, only a low number of concrete metrics (in different degree of detail), are described by existing EA management literature. Accounting for the idea of using best-practices in our domain we present an organized collection of metrics, which usage we observed in industry. Further, by using our description structure and accounting for the goal-question-metric idea, we provide concrete metric recommendations with respect to given EA management goals and concerns.

4. Accounting for the BEAMS idea of supporting a holistic life-cycle management of a minimal and organization-specific EA management function for a given enterprise (by using corresponding best-practices in the domain) we propose a corresponding method and a BEAMS extension supporting the holistic life-cycle management of organization-specific metrics under consolidation of metric best-practices in the domain. Existing literature confirms the importance of specialized tool support for EA management functions. Nevertheless, to our knowledge, a corresponding software support for our method is currently not available. In addition, practitioners confirm to apply currently Excel spread sheets or BI-based solutions for the purpose of implementing an EA management metrics management, which leads to different problems based on the separation of qualitative and quantitative models. Hence, we extend the commercial EA management tool Tricia by a new domain-specific language, providing a minimal number of language concepts required to implement the metric best-practices from our catalog to allow holistic life-cycle management for both—quantitative and qualitative models within one tool.

1.1. Research Questions

According to our solution description, we define the subsequent hypothesis for our research:

> ***Research hypothesis:*** Our software supported method allows organizations with established EA management functions to implement a holistic life-cycle management for organization-specific metrics.

In line with Gläser et al. [GL06], answering research questions must add new knowledge to the existing body of knowledge. Since research questions guide the evaluation process of the results, we explicitly define the following research questions with respect to our research hypothesis:

> ***Research question 1:*** What are possible usage scenarios for quantitative models (metrics) in the domain of EA management?

To answer this question, we analyze existing literature in our domain and take the recommendations from related management fields, e.g. IT controlling, and enterprise management into account. Thereby, we identify concrete usage scenarios for metrics in our domain and provide a consolidation of these scenarios by taking a systemic view on an enterprise.

> ***Research question 2:*** What is an appropriate terminology for quantitative models (metrics) in the domain of EA management?

As existing literature confirms, a multitude of different terms is used by different authors (cf. Section 3.2). To establish a clear and comprehensive terminology for our research, we study the terms used in literature. Under consideration of the previously identified usage scenarios, we propose a comprehensive terminology base for our domain.

> ***Research question 3:*** What is a generic and minimalistic metric management fact sheet for our domain?

Since metric management fact sheets (MMFSs) proved to be an important part of the metric management process in related management fields, e.g. [Kü10, Kü13, Pa07, NAK02], we design and establish a generic MMFS tailored to the domain of EA management on the one hand, and ensuring a comprehensive metric description on the other hand. Further, we account for a minimal number of metric description elements in this fact sheet, which are indispensable to achieve our goal. In addition, the design of the fact sheet supports user-organization in the organization-specific configuration and instantiation of metrics to their specific contexts.

> ***Research question 4:*** What are EA management metric best-practices used in industry?

As patterns [Er10] and building-blocks [Bu11, Sc11] proved to provide valuable support for experts in the field of EA management, we observe metric best practices in German industry and document these observations using our metric management fact sheet. Further, we create an organized collection of the observed metric best-practices to support experts in the timely identification of recommended metric best-practices based on corresponding EA management goals and concerns.

> ***Research question 5:*** What is a holistic life-cycle management method for metrics tailored to a specific organizational-context in the domain of EA management?

As literature confirms, cf. Section 2.1.3, existing EA management frameworks account for a holistic life-cycle management of qualitative EA models by employing specific methods. Nevertheless, a method for the holistic life-cycle management of EA quantitative models is missing, cf. Section 2.3.8. Hence, in this thesis we present a corresponding metric method and additionally, we integrate this method into the BEAMS framework developed at our chair. By this integration, the framework provides holistic life-cycle management methods for both—qualitative and quantitative EA models.

> ***Research question 6:*** What are risks and suggested countermeasures associated with the usage of EA management metrics?

To answer this question, we consolidate risks and suggested countermeasures associated with the usage of metrics in related management disciplines. Thereafter, based on the feedback of a survey with enterprise architects from German industry, we propose a (prioritized) list of

26 risk and recommended countermeasures for our domain. In addition, we incorporate this results in the design of our metric management method.

> ***Research question 7:*** What is a suitable and minimalistic design of a domain-specific model query language for the implementation of the catalog's metrics?

Based on a set of concrete requirements for the language design according to literature and to the feedback gained in collaboration with industry experts, as well as by considering the knowledge in the field of software engineering, we propose a concrete design and a prototypical implementation of a corresponding domain-specific query language.

> ***Research question 8:*** What are requirements towards an integrated software support for the proposed holistic metric life-cycle management method?

Based on the design of the created constructs, models, and methods as answers to the previous research questions and considering the input gained in discussions with experts from industry, we define concrete requirements for the intended software solution, which are to be met in the implementation phase of our research. In addition, the validity of these requirements is part of the evaluation of the prototype.

1.2. Research Method

Many different methods to conduct research have been introduced and applied in the field of *information systems* (IS). The development of a software-supported holistic metric life-cycle management method in the EA management domain results in a new IS design artifact. To understand, execute, and evaluate our research in a scientific appropriate way, we adhere to the seven guidelines for IS design science proposed by Hevner et al. [He04]. Originating from the field of engineering, design science applies a problem-solving oriented process. According to Hevner et al. [He04], this process adheres to the principle that knowledge and understanding of a design problem and its solutions are acquired in the building and application of an artifact. The design-oriented research creates the following four distinct types of artifacts:

- *Constructs* - providing a language to describe problems and related solutions.
- *Models* - describing reality and enabling abstraction.
- *Methods* - defining processes which in term are guiding the problem solution.
- *Instantiations* - demonstrating that the constructs, models, and methods can be implemented in a working system.

The artifact we present in our research originates from two basic sources: scientific literature as well as industry expert surveys, interviews, and discussion rounds. Both types of sources enable the collection of data for the creation and validation of the artifact. Our artifact provides a clear definition of the applied terminology and identified usage scenarios in terms of constructs. According to the terminology of Hevner et al. [He04], both—the proposed MMFS structure and the metric catalog represent models, whereas the proposed holistic metric life-cycle management method represents a method. Further, the developed software prototype can be considered as an instantiation of the method. Hence, we follow the "seven design science research guidelines" of Hevner et al. [He04], as described subsequently.

Problem relevance The absence of a holistic life-cycle metric management method as well as an integrated software support by a corresponding EA management tool, is an important and relevant problem in this domain as already pointed out. To emphasize the significance of this research goal, as well as to describe the related blind-spots in research literature, we conducted an exploratory literature review and presented the results, together with a high-level solution design description, at the PhD Track of the EDOC conference in 2011 [Mo11]. Later we extended our literature review also to the fields of IT management and enterprise management to ensure that all relevant knowledge is taken into account for answering our research questions. As the evaluation of our expert surveys and interviews shows (cf. Chapter 4), the problem is relevant for enterprise architects, as well as business users and IT managers.

Design as an artifact Our software prototype is a viable IS artifact, combining several distinct types of created sub-artifacts, e.g. constructs, models, and methods. These artifacts guide enterprise architects and business-users how to define, configure, instantiate, implement, and manage organization-specific EA management metrics tailored to their goals and needs, using a generic metric management fact sheet under consideration of known metric best-practices.

Research contribution The contributions of this thesis are described in Table 1.1.

Research rigor By taking the existing knowledge in our domain into account, talking to, and learning from industry experts, using meta-modeling techniques, providing formal description for EA management metrics, and by applying well-known software-engineering patterns and constructs for the implementation of our solution, this thesis preserves rigor.

Design as a search process The IS artifact, as well as its sub artifacts developed in this thesis, have gone through several iteration cycles. The initial solution was outlined and published firstly in [Mo11]. Then it was refined as described in [Ma12c, Ma12b]. Further, the method was prototypically implemented using the gained feedback and experience from [MRM13]. Based on the feedback from the application of the first version of the prototype in two independent research projects (cf. Section 5.2.1), the prototype was improved as described by Reschenhofer [Re13] and in Chapter 5.

Communication of research The final version of our metric management method is published in this thesis to make it available to both, management and technology audiences in an academic as well as industry environment. The software prototype is described in detail in this work and all of our cited publications are accepted at conferences or scientific workshops after passing a (double) blind peer-review process.

Design evaluation Every created sub-artifact of the presented solution is evaluated separately as described in the related evaluation sections – Chapter 4 for the evaluation of the conceptual part of this thesis and Chapter 6 for the evaluation of the software engineering part. The evaluation of the software prototype is done based on the subjective feedback of experts, who used our prototype during its application in a German public bank as an alternative to an existing software solution developed by the bank for the same purpose. Further, for the evaluation of our prototype, we define an alternative hypothesis with the objective to falsify it and hence, to prove the validity of our research hypothesis. The evaluation stresses the design of our solution with respect to the validity and completeness of our predefined architectural requirements and in particular, the predefined requirements of the MxL design. Furthermore, we investigate the advantages and disadvantages provided by our solution compared to the solution used by the bank (cf. Chapter 6). In addition, our method and provided integrated software support are designed to support the holistic life-cycle management of EA management metrics. During the evaluation of the prototype we could evaluate the definition, instantiation, implementation and prototypical usage of metrics. Nevertheless, due to time limitations and due to a missing commercial version of our software, it was not possible to evaluate the management of the developed metrics, i.e., we could not observe changes to the defined metrics over time, or the deletion of no longer required metrics.

1.3. Structure of the Thesis

Figure 1.1 illustrates the outline of this thesis. Thereby, the work is organized into main sections - a conceptual part and a software engineering part. This ensures a proper separation of the concept development and the software engineering phases in our research process.

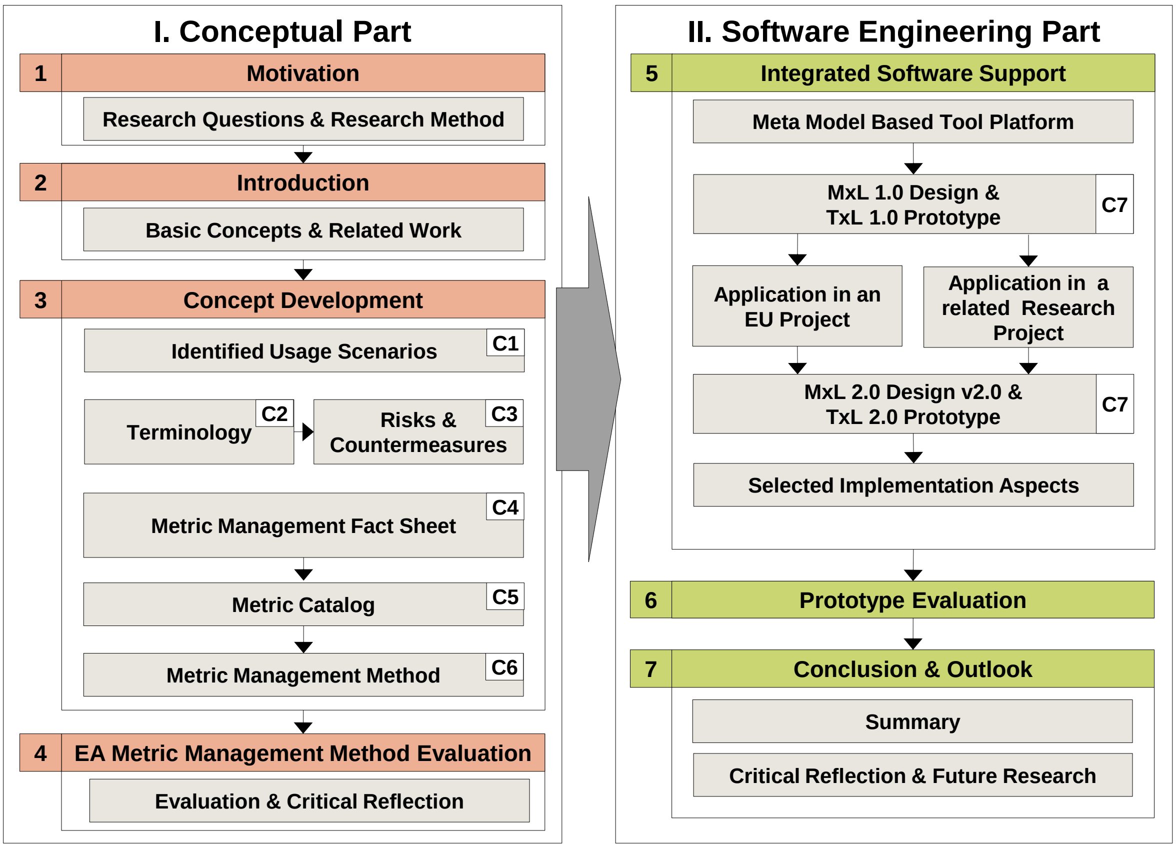


Figure 1.1.: Outline of the thesis linked to the sub-contributions, cf. Table 1.1

CHAPTER 2

Introduction and Related Work

2.1. Introduction

In this Chapter, we present relevant foundations and concepts from the related literature for our research. Therefore, we introduce the concept of enterprise architectures in Section 2.1.1. Consequently, in Section 2.1.2, we present relevant foundations from the domain of EA management. In the subsequent Section 2.1.3, we introduce the foundations of the BEAMS framework developed at our chair. As a part of our contribution, we aim the integration of a holistic metric life-cycle management method in BEAMS. Subsequently, in Section 2.2, we present fundamental concepts related to the usage of metrics in management disciplines. Further, in Section 2.3, we describe the process for the identification of related literature for our research and the results of its application. Additionally, we present seven selected approaches from the related EA management literature. These works provide valuable concepts, which we integrate into our solution's design.

2.1.1. Enterprise Architecture

Rooted back in the domain of IS architecture [SZ92], EA management represents a commonly accepted discipline to cope with the complexity of change in and the problem of missing alignment between business and *information technology* (IT) [LLO93, RWR06, ARW08b, Sc09, Ke07]. According to Buckl et al. [BS11], EA management has been subject of interest for academia, practitioner and consultants for over three decades. However, according to Schönherr's literature review of over 126 publications in this area [Sc08], a common definition of the term enterprise architecture is still missing. For this work and also in line with the majority of the existing literature, we define the term enterprise architecture according to the ISO Standard 42010 [In07, page 3] as:

Definition: Enterprise architecture
Enterprise architecture is the fundamental conception of the organization in its environment, embodied in its elements, their relationships to each other and to its environment, and the principles guiding its design and evolution.

As this definition suggests, enterprise architecture considers the system (enterprise) from a holistic perspective. According to Wittenburg [Wi07b], the EA covers all elements of an enterprise from business and organizational via application and information to infrastructure and data aspects. Among others, the following two important advantages are offered by enterprise architecture(cf. [ARW08a, Th09]):

- foster communication by a common language for multidisciplinary stakeholders, and
- gather information from differing sources and provision of consistent decision base.

Figure 2.1 illustrates the overall structure of the EA according to Buckl [Bu11] and Schweda [Sc11].

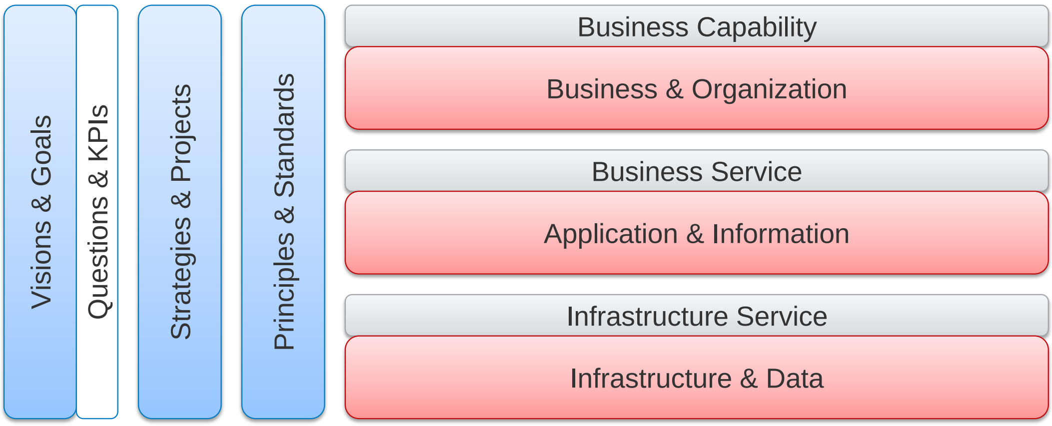


Figure 2.1.: Overall structure of the enterprise architectureaccording to [Bu11, Sc11]

The EA consists of the three *architectural layers* - **Business & Organization**, **Application & Information**, and **Infrastructure & Data**. These architectural layers reflect the overall business-to-infrastructure structure of the organizations' architecture. It ranges from logical concepts on the business and organization level (e.g. products, organizational units, business processes), which are independent of the technical realization, over application level concepts, that describe the IT realization of the logical concepts (e.g. business applications and interfaces), to infrastructure concepts (e.g. logical servers).

In addition, the EA consists of three *abstraction layers* - **Business Capability**, **Business Service**, and **Infrastructure Service**. These abstraction layers complement each of the architectural layers with a customer-oriented perspective. They describe the EA concepts on the corresponding architectural layer in an abstract way. Thereby, these abstraction layers focus on the provided functionalities, whereas details of the actual implementation of the functionalities are hidden (black-box perspective). The architectural and abstraction layers build the so-called *static part of the EA structure*.

The EA is complemented by the following four cross-cutting aspects - **Visions & Goals**, **Questions & Key Performance Indicators (KPIs)**, **Strategies & Projects**, and **Principles & Standards**. These cross-cutting aspects cover concepts that are not directly part of the static EA structure but may be linked to any element in a layer in different ways, e.g. linking EA goals via metrics to concrete EA elements of different architectural and abstraction layers.

To sum up, "the EA describes the current state of the enterprise (descriptive aspect) and makes prescriptions for its planned and target states (normative aspect)" [Sc11]. This different architecture plans allow organizations to document and control changes on every layer and cross-cutting-aspect of the overall EA structure.

2.1.2. Enterprise Architecture Management

According to Buckl [Bu11], the multitude of changes in organizations ties in with a growing internal complexity (i.e., the growing number of diversity of EA elements and the number of relationships between them) of the socio-technical system of the enterprise [Bu11]. This development motivates many organizations to introduce a so-called EA management function. Thereby, the organizations usually seek to realize the following benefits associated with EA management: (cf. [RWR06, Ke07, ARW08b, Sc09, Bu11, Sc11]):

- consistent strategic IT planning,
- increased business/IT alignment,
- business process optimization, and
- architectural guidance for (change) projects.

According to Schönherr [Sc08] and in analogy to the term EA, the term EA management also does not have an unique definition. For our work, we define EA management as a general management function targeting the EA. More precisely, we stick to the definition of Schweda [Sc11]:

> **Definition: EA management function**
> The EA management function in an enterprise documents, analyzes, plans, and enacts the EA.

As part of the term EA management, the word *management* generally refers "to the process of assembling and using resources - *human*, *financial*, *material*, and *information* - in a goal directed manner to accomplish tasks in an organization" [BP00]. Concerned with the present, the expected, and the desired future [Dr06], management functions are usually described by a *planning*, *leading*, *organizing*, and *controlling* dimension [BP00]. In EA management, the objective-directed character is realized through the definition and pursuit of specific EA management goals. Based on the specific goals and business environment of different organizations, EA management frameworks focus on the development of a so-called *organization-specific* EA management function, tailored to the needs of a given organization and the involved stakeholders [Th09, Bu11, Sc11].

Regarded as the basic purpose of any EA management initiative, each individual EA management goal represents an abstract objective ideally supporting at least one business goal [Bu10]. Buckl et al. also provide a list of ten common EA management goals, e.g. *increase homogeneity* or *provide transparency*, which are considered as part of their EA management framework *BEAMS* [Bu11, Sc11].

According to Buckl et al. [BS11], 22 EA management frameworks existed in 2011. One of these frameworks - the *Building Blocks for Enterprise Architecture Management Solutions* (BEAMS) is developed at our chair. With respect to the research goal of this thesis, the BEAMS framework supports the development of lean and organization-specific EA management functions according to the goals and concerns of involved stakeholders in a given organization. Nevertheless, a corresponding development method for quantitative models (metrics) is not addressed by the framework. Hence, to close this gap, we target the development and the integration of a holistic life-cycle management method for organization-specific EA management metrics within the BEAMS framework (cf. Section 3.6). Consequently, the subsequent section introduces the foundations of the BEAMS framework, which we integrate in our solution.

2.1.3. Building Blocks for Enterprise Architecture Management Solutions

Inspired by the concept of EA management patterns (cf. Buckl et al. [Bu08]), Buckl [Bu11] and Schweda [Sc11] presented their building-block based approach for EA management solutions (BEAMS) in 2011. This approach provides a *method base* to capture best-practice knowledge on EA descriptions, so-called *building blocks*. The method base is developed and evolved by the iterative EA management activity framework presented by the authors as depicted in Fugure 2.2. Thereby, the framework consists of the following four activities:

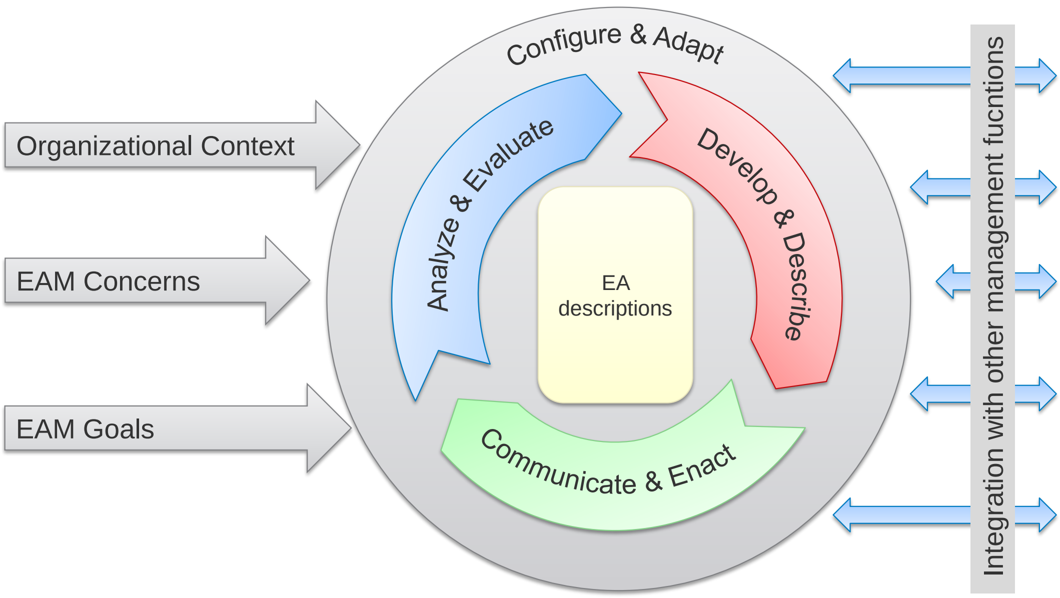


Figure 2.2.: BEAMS activity framework according to Buckl [Bu11] and Schweda [Sc11]

- *Develop & describe* is concerned with the development of EA descriptions. Thereby, different architectural states are described - *current*, *planed*, and *target* EA states. During this activity, the concerns of the involved stakeholders are documented and a concrete roadmap describing the evolution from the current to the target state is developed.
- *Communicate & enact* is concerned with the communication and enactment of the developed architectural descriptions from the previous activity in the related management areas. Thereby, different strategies for the implementation of this activity can be applied depending on the given organizational culture, e.g. provision of benefits for compliance or financial penalties for non-compliance with respect to the predefined target states.
- *Analyze & evaluate* is concerned with making different architectural states comparable to support subsequent decision making. If a reached plan state does not match the expected results, countermeasures have to be defined and implemented to achieve the predefined stakeholder goals.
- *Configure & adapt* is concerned with the definition of the scope, vision, and proper empowerment of the EA management program in a given organization. Thereby, before starting the EA management program, organizations have to clearly define the pursued goals, concerns, and related problems. This activity is also important to redefine the EA management program at any point in time as a response to changes, e.g. changing organizational context, goals, or market environment.

This EA management activity framework provides a development method for organization-specific EA management functions on the one hand, and an administrative method for the evolution of the underlying method base on the other hand. Figure 2.3 describes the main BEAMS contribution - the development method for organization-specific EA management functions. The steps of this method are described as follows:

- *Characterize situation* - in this step, enterprise architects and the involved stakeholders collect the required input for the EA management function. Thereby, the given organizational context as well as the problems of the relevant stakeholders have to be documented. According to the authors, a problem has to be decomposed in goals and concerns, e.g. the problem "the degree of heterogeneous technologies in our application landscape is too high" can be decomposed in the goal "increase homogeneity" and the concern "application landscape".
- *Configure BB* - having collected the required input from the previous step, enterprise architects use the EA descriptions provided by the method base to define an organization-specific EA management function. Thereby, the building blocks selected from the method base have to be configured and interconnected. The BEAMS framework distinguishes between following two basic types of building blocks - *Method Building Blocks* (MBBs) and *Language Building Blocks* (LBBs). A MMB describes which stakeholder has to perform which tasks to address a specific problem in a given organizational context. A LBB is concerned with the EA information required to perform the task described by an MBB and with the visualization of this data. Further, LBBs are distinguished in following two subtypes, *Information Building Blocks* (IBBs) and *Viewpoint Building Blocks* (VBBs). For more information regarding MBB we refer to the PhD thesis of Buckl [Bu11] and regarding LBBs to the PhD thesis of Schweda [Sc11].

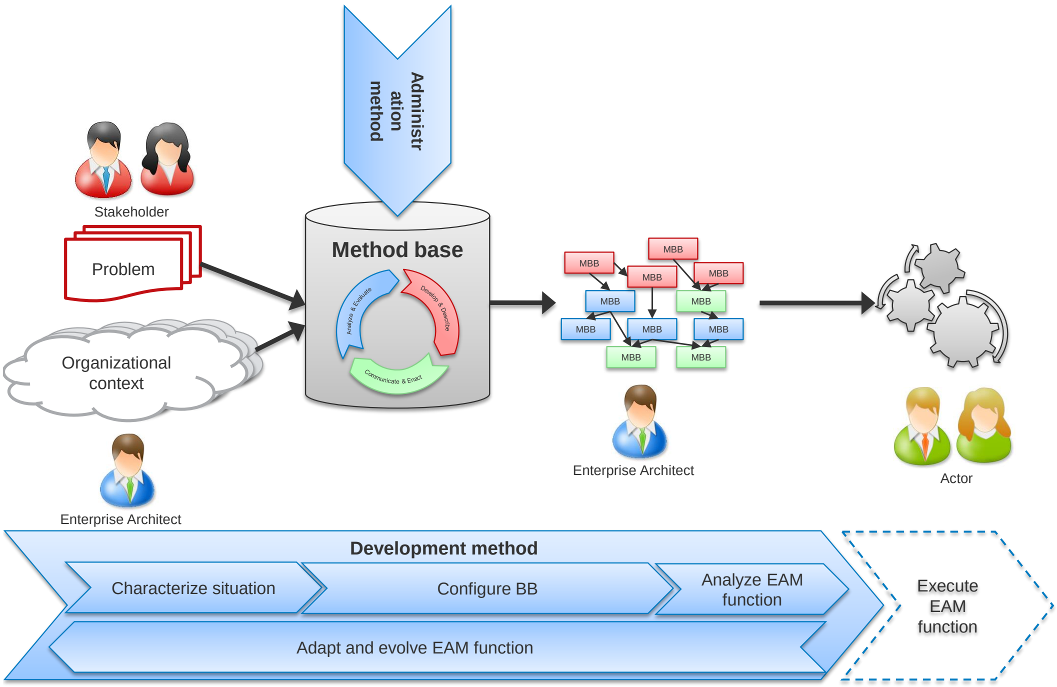


Figure 2.3.: The BEAMS framework according to [Bu11, Sc11]

- *Analyze EAM function* - after configuring the building blocks, the enterprise architects have to ensure in this step, that the developed EA management function meets the expectations of the stakeholders. The BEAMS framework provides different qualitative techniques how to perform this analysis. Once the analysis is finished, the organization can adopt and execute the developed EA management function.

- *Adapt and evolve EAM function* - This step is concerned with the management of the evolution of the EA management function. As already mentioned above, the EA management function may have to be adapted to a changing organizational environment, e.g. new organizational goals, business strategy, or changed stakeholder concerns. For additional information about BEAMS we refer to the original sources [Bu11, Sc11].

To sum up, the BEAMS framework supports the organization-specific development of EA management functions by using building blocks representing best-practice EA solutions. The framework helps organizations to keep the focus on the relevant EA parts with respect to the goals and problems of the involved stakeholders and thus, to ensure a lean EA management scope. In addition, it supports the managed evolution of established programs as a response to organizational changes.

2.2. Fundamental Concepts for Metrics in Management

Enterprise architectures require specific management functions to guide their evolution. According to Black et al. [BP00], management functions are usually described by the following four activities:

- *Planning* - the ability to make decisions about actions needed to be done in the future to achieve given goals based on the current state and the expectations of future states. Usually, plans are done for the staff. The amount of the staff, the granularity of the plans and their time frames can differ from *strategic* to *operative* planning level.
- *Leading* - the ability to motivate, influence and guide others to achieve given goals.
- *Organizing* - the ability to adequately use, combine, and utilize resources to achieve given goals at different organizational layers, e.g. organizational units and teams.
- *Controlling* - the ability to monitor and evaluate the degree of the current goal achievement under the responsibility of a manager. There are many different approaches for controlling described in literature - *definition of standards in advance*, *evaluation of completed work*, or *performance measurement.*

Put in other words, management is concerned with the current, the expected and the desired future [Dr10] of the management subject. According to [BP00], persons performing management are referred to as *managers* and they have to fulfill the following three types of roles:

- *decision roles*, e.g. ensuring they have all related input required to make long or short-term decisions,
- *information roles*, e.g. communicating with people in and outside of their area, and
- *interpersonal roles*, e.g. ceremonial activities (promotion talks), demonstrating leadership and extending their network outside of their usual area.

According to [Kü10, SRS10, NAK02], managers require quantitative facts in a highly concentrated form as a decision base. For this purpose, they usually work together with *controllers*, which in turn are responsible for collecting and providing appropriate data to the managers as basis for decision making. According to Kütz [Kü10], controllers typically focus on the definition and implementation of metrics and metric systems. In line with this idea, Probst [Pr12] states, that metrics are indispensable controlling instruments for every organization. As depicted in Section 3.2, several definitions of the term metric exist. A common definition of this term from management literature is given by Siegwart et al. in [SRS10], where a metric is defined as a "[...] number, providing economically reasonable information in concentrated form". Furthermore, a metric can be understood as "a compression of quantitative information" [SRS10]. The authors distinguish between two types of metrics in enterprise management, as depicted in Figure 2.4. *Absolute numbers* are metrics which are used independently by other metrics. Absolute numbers provide an immediate description of the current state, process, or phenomenon they are designed for to quantify. Thereby, the authors distinguish the following four distinct subtypes of an absolute number:

- a *single number*, e.g. *revenue*,
- a *sum*, e.g. *balance sheets*,
- a *difference*, e.g. *working capital*, and
- an *average*, e.g. *average stock of inventory.*

Secondly, *ratios* are characterized by consideration of two interrelated factors. Thereby, a ratio is expressed by a factor or %. The following three subtypes of ratios are distinguished:

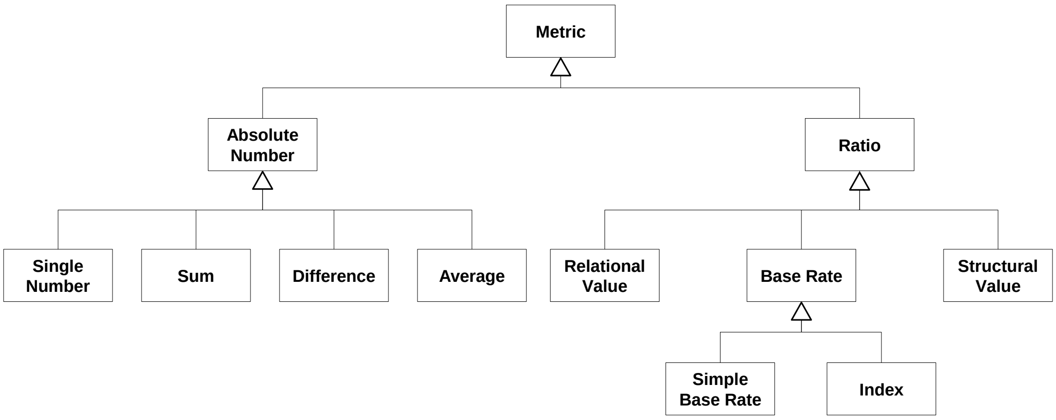


Figure 2.4.: Types of metrics according to Siegwart et al. [SRS10]

- A *structural value* is used when a "whole object" is decomposed into parts, and the parts are set in relationship to the whole, e.g. *balance analysis*.
- A *relational value* is used to set two objects with the same numeric scale (cf. von Dobschütz [Do00]), e.g. absolute or relative scale, but from different types (in sense of objects types, e.g. employee and revenue) into relationship, e.g. *revenue per employee*. The authors emphasize that relational values are the most important type of metrics in the area of enterprise controlling. Nevertheless, every relational value must be empirically evaluated in terms of correlation significance and meaningfulness. For example, the relational value *revenue per square meter sales area* is a useful metric for an enterprise in the trading industry. According to Siegwart et al. [SRS10], for the same enterprise, the relational value *revenue per square meter production area* makes no sense.
- *Base rates*. Thereby, the authors distinguish between two types of base rates. On the one hand, *simple base rates* are used for evolution analysis of concrete enterprise data, cf. Table 2.1. On the other hand, an *index* allows the evaluation analysis of several objectively related series of enterprise data, e.g. *price index*.

Year	2005	2006	2007	2008	2009
Net Income (in 1000€)	37 300	37 900	38 900	39 300	39 900
Base Rates	100	101.6	104.3	105.4	107

Table 2.1.: Example of base rates usage (based on Siegwart et al. [SRS10])

The usage of metrics enables the time comparison of quantitative information. For instance, according to Siegwart et al. [SRS10], the comparison between the current *return on equity* (ROE) and the ROE from the last years (cf. Figure 2.5) is considered as an indispensable business information. Time comparison of metrics allows:

1. an overall picture of the financial situation of an enterprise,
2. the understanding of the evolution of an enterprise, and

$$Return\ on\ equity\ (ROE) = \frac{Net\ income}{Shareholder\ equity}$$

Figure 2.5.: ROE formula according to [SRS10]

3. the early identification of negative developments and thus, supports the ability to timely define and implement corresponding countermeasures.

In line with existing controlling literature [KN91, NAK02, Kü10, KA10, Pr12], metrics (in German *Kennzahlen*) are required by organizations to support internal steering and controlling, reporting, early warnings, and problem identification. For specific management perspectives, there are well known and widely accepted metrics, e.g. :

- *Financial perspective* - earnings before taxes, return on investment, cash ratio,
- *Customer perspective* - customer acquisition rate, profit margin, break-even point,
- *Process perspective* - cost performance index, throughput time, process cost rate, and
- *Human resource perspective* - personnel cost ratio, overtime quota, employee satisfaction index.

Calculating metrics on a regular basis from these different perspectives allows a balanced analysis of the efficiency and effectiveness of an organization's performance and supports the identification of performance improvement potentials, as presented by Kaplan et al. in their *Balanced Scorecard Approach* (BSC) [KN91]. Thereby, the idea of combining financial and non-financial metrics in the field of enterprise controlling is not new. According to Siegwart et al. [SRS10], the most important innovation in the BSC concept is the usage of these distinct perspectives to support cause-effect relationship analysis.

According to Kütz [Kü10], the definition of a metric requires a clear definition of a so-called "steering object" and related "controlling goals". Thereby, the definition of the steering object must exactly describe which parts of the organization (system) belong to the scope of the controlling initiative and which do not. Afterwards, a responsible manager has to be assigned. Thereafter, both—the controller and the responsible manager have to collaboratively refine the controlling goals. Thereby, in analogy to the different states of a given EA (cf. Section 2.1.3), the goal descriptions should contain the following three states:

- the *current state* of the steering object,
- one or more *planned states* the steering object must pass through its evolution, and
- a *targeted state* the steering object should be transformed to.

It is not necessary, that the target (and planned) state(s) must differ from the current state, i.e., the controlling goal might be to ensure, that a given steering object stays stable over time. However, if these states differ, the goal definitions must provide a clear description of the time required to achieve the target state and the plan of transformation. After defining the different steering object states, controllers and managers have to define a metric system to measure the steering object in order to ensure continuous measurement and deviations analysis to the predefined planned and target states. A set of possible countermeasures is to be defined, which can be executed to influence the steering object and thus, to influence the values of the metrics. These six elements - a steering object, controlling goals (containing the current, as well as planned and/or target states), the defined metric system, the defined countermeasures and the ability to analyze deviations between different states of the steering object are defined as the *basic controlling loop* (cf. Figure 2.6). The most important four

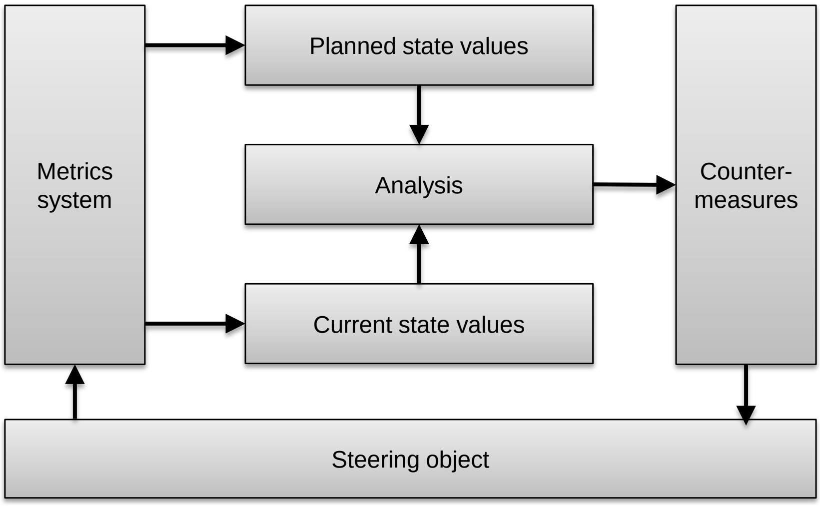


Figure 2.6.: The basic controlling loop according to Kütz [Kü10]

conditions for a successful controlling are [Kü10]:

1. The steering object has to be clearly described and delimited from its environment.
2. A clear definition of the responsible manager for the steering object is indispensable.
3. The steering object and the developed metric system are to be understood as functions of time. Both can change/evolve only when time passes by. It is necessary to define current, planned, and target states to ensure continuous measurement of the goal achievement at any point in time. Additionally, these states must be comparable with each other.
4. Both—managers and controllers must have a shared understanding of how the steering object can be influenced by the predefined countermeasures in order to meet the planned and target states.

According to [Kü10, FGM07], a metric always represents a model of the reality. The correctness and usefulness of such models are usually determined in their practical application. Typically, controllers require more than only one single metric in order to measure the degree of achievement of a given controlling goal. The literature refers to such sets of metrics as *metric systems*. Looking on a metric system from a mathematical viewpoint, it describes the steering object by (time dependent) vectors with respect to the used metric system as depicted in Figure 2.7. Thereby, every steering object must have an initial current state value vector and for each planned (and target) state a corresponding value vector is required. Equipped

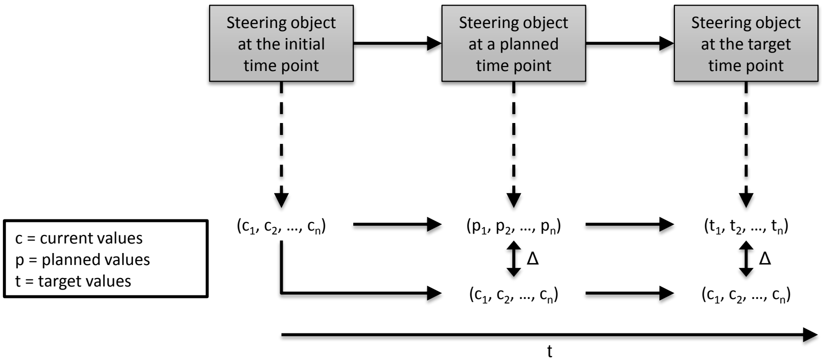


Figure 2.7.: Steering object and it's controlling value vectors according to Kütz [Kü10]

with this information, controllers can calculate deviations (deltas) and suggest concrete countermeasures to the managers to ensure the achievement of the predefined controlling goals.

Although, metrics and metric systems are well understood and widely accepted in industry and academia, they cannot answer all question regarding the steering object and they cannot replace human decision makers [SRS10, Kü10, Pr12].

2.3. Related Work

According to our research method (cf. Section 1.3), a comprehensive knowledge on the existing literature is required to ensure the consideration of relevant knowledge as basis for our research, and to justify the provision of an unique new research contribution. For this purpose, we apply the *effective literature review process* by Levy and Ellis [LE06] as illustrated in Figure 2.8. This

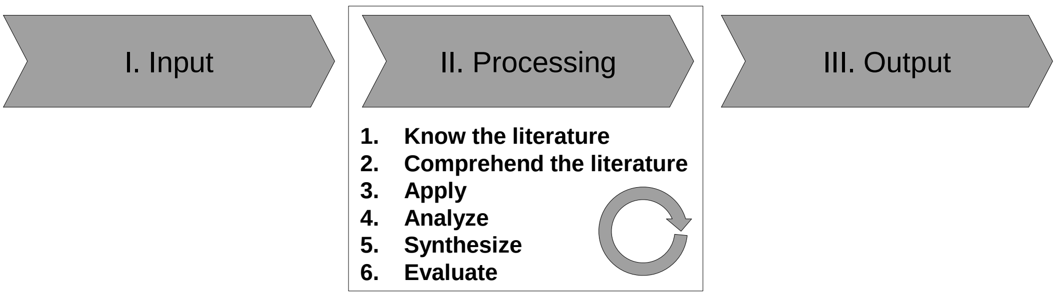


Figure 2.8.: An effective literature review process according to Levy et al. [LE06]

process is developed for the domain of IS, however, according to the authors, it can be applied in social and behavioral science as well. It consist of the three phases:

Input - as described in Chapter 1, based on the increasing importance of EA management and the growing demands for quantitative models in the last years, there is also an increasing number of research and practitioner publications in the field as our literature review confirms. To ensure the identification of all relevant literature sources, we focus on contributions in leading journals and also on electronic resources for library services, as suggested in [LE06]. For this purpose, we use the following three search engines: *Google*, *Google Scholar* and the *search engine of the TU Munich's library*, which grants us access to the three publications databases *IEEE*, *ACM* and *CiteSeer*. Thereby, our review accounts for the following three techniques according to [LE06]:

Keyword search - the literature search is performed using the following search terms: "enterprise architecture", "EA", "enterprise architecture management", "EA management", "EAM", "quantitative model", "metric", "indicator", "performance indicator", "KPI", "measurement", "risk", "issue", "threat", "drawback", "limitation", and "EA benefits", as well as different combinations between these terms. The search is applied in literature from the fields of EA management, IT management and enterprise management. We also performed the same search for the German translations of the keywords.

Backward search - the backward search is conducted in three steps. Firstly, a *backward references search*, concerned with the review of the references of the articles as identified by the keyword search is performed. Secondly, a *backward authors search* concerned with the review of previously published works of the authors as identified by the keyword search is performed. Lastly, a *previously used keywords search*, concerned with the identification of additional key words based on the findings from literature is performed.

Forward search - this search is conducted in two steps. Firstly, a *forward references search*, concerned with the review and identification of relevant sources citing the findings from the previous steps is performed. Secondly, a *forward authors search* is performed. Thereby, based on the current findings, we review the publications lists of the authors with the goal, to identify relevant publications beyond the publication time point of their work as known to us.

Processing - after identifying all relevant literature sources as described above, the sources are to be analyzed. With respect to our research questions (cf. Sections 1.1), we concentrate during this processing phase in particular on the relevance of each finding regarding following five aspects:

1. Identification of existing methods, approaches and frameworks in the domain of EA management targeting the definition, development, and holistic life-cycle management of EA management metrics.
2. Identification of usage scenarios for the usage of quantitative models (metrics) in the field of EA management.
3. Establishing a consistent terminology base for this thesis.
4. Identification of typical risks, problems, and recommended countermeasures associated with the usage of metrics in management disciplines.
5. Identification of typical elements and structures for the documentation of metrics in different management fields.

Output - based on the results from the processing phase, we identified 40 sources originating from the fields of EA management (cf. Table 2.2), IT management (cf. Table 2.3) and enterprise management (cf. Table 2.4), which we consider as relevant for our research.

Furthermore, we use these works as a pool of related sources, which we consider during the design of our conceptual solution, cf. Chapter 3. Thereby, for each of our research questions (cf. Section 1.1), we investigate all of these literature sources with the goal to account for and incorporate the existing knowledge in our work.

First author	**Title**	**Key**	**Document type**
Addicks, Jan Stefan	Bewertung betrieblicher Anwendungen im Kontext ihrer Unternehmensarchitektur	[Ad10]	PhD thesis
Aier, Stephan	Understanding Enterprise Architecture Management Design – An Empirical Analysis	[AGW11]	Conference paper
Buckl, Sabine	State of the Art in Enterprise Architecture Management 2009	[Bu09]	Technical report
Buschle, Markus	The Enterprise Architecture Analysis Tool — Support for the Predictive, Probabilistic Architecture Modeling Framework	[BJS13]	Conference paper
Feldschmid, Andreas	Konzeption und prototypische Implementierung eines Steuerungscockpits im Kontext des Managements von Unternehmensarchitektur	[Fe09a]	Master's thesis
Gringel, Philipp	Metriken zur Bewertung von Anwendungslandschaften	[Gr09]	Master's thesis
Johnson, Pontus	P2AMF: Predictive, Probabilistic Architecture Modeling Framework	[Jo13a]	Conference paper
Kaisler, Stephen H.	Enterprise Architecting: Critical Problems	[KAV05]	Conference paper
Lankes, Josef	Metrics for Application Landscapes – Status Quo, Development, and a Case Study	[La08]	PhD thesis
Niemi, Eetu	Enterprise architecture benefits: Perceptions from literature and practice	[Ni08]	Conference paper
The Open Group	TOGAF, Version 9.1	[Th09]	White paper
Plessius, Henk	On the Categorization and Measurability of Enterprise Architecture Benefits with the Enterprise Architecture Value Framework	[PSP12]	Conference paper
Stutz, Matthias	Kennzahlen für Unternehmensarchitekturen: Entwicklung einer Methode zum Aufbau eines Kennzahlensystems für die wertorientierte Steuerung der Veränderung von Unternehmensarchitekturen	[St09b]	PhD thesis

First author	Title	Key	Document type
Strecker, Stefan	MetricM: a modeling method in support of the reflective design and use of performance measurement systems	[St12]	Journal article
Winter, Robert	Analysis and Application Scenarios of Enterprise Architecture – An Exploratory Study (Reprint)	[Wi07a]	Journal article

Table 2.2.: Output overview from EA management literature

First author	Title	Key	Document type
Basili,Victor R.	The Goal Question Metric Approach	[BCR94]	Book
Franceschini, Fiorenzo	Management by Measurement: Designing Key Indicators and Performance Measurement Systems	[FGM07]	Book
Kütz, Martin	Kennzahlen in der IT. Werkzeuge für Controlling und Management	[Kü10]	Book
Kütz, Martin	IT-Controlling für die Praxis – Konzepte und Methoden	[Kü13]	Book
Steinberg, Randy A.	Measuring ITIL: Measuring, Reporting and Modeling – the IT Service Management Metrics That Matter Most to IT Senior Executives	[St06]	Book

Table 2.3.: Output overview from IT management literature

First author	Title	Key	Document type
Bird, Sheila M.	Performance indicators: good, bad, and ugly	[Bi05]	Journal article
Bourne, Mike	Implementing performance measurement systems: a literature review	[Bo03]	Journal article
Black, Stuart	Management: Meeting New Challenges	[BP00]	Book
Eckerson, Wayne W.	Performance dashboards: measuring, monitoring, and managing your business	[Ec10]	Book
Eccles, Robert G.	The performance measurement manifesto	[Ec90]	Journal article
Hauser, John	Metrics: you are what you measure!	[HK98]	Journal article
Kaplan,Robert S.	The Balanced Scorecard – Measures That Drive Performance	[KN91]	Journal article
Kotter, John P.	Leading change: Why transformation efforts fail	[Ko95]	Journal article
Lawson, Raef	Scorecard best practices: design, implementation, and evaluation	[LDH08]	Book

First author	Title	Key	Document type
Lebas, Michel	A conceptual and operational delineation of performance	[LE02]	Journal article
Neely, Andy D.	The performance prism: the scorecard for measuring and managing business success	[NAK02]	Book
Neely, Andy D.	Business performance measurement: theory and practice	[Ne02]	Book
Neely, Andy D.	The performance measurement revolution: why now and what next?	[Ne99]	Journal article
Neely, Andy D.	Performance measurement system design: a literature review and research agenda	[NGP95]	Journal article
Parmenter, David	Key performance indicators (KPI): developing, implementing, and using winning KPIs	[Pa10]	Book
Perrin, Burt	Effective use and misuse of performance measurement	[Pe98]	Journal article
Popova, Viara	Modeling organizational performance indicators	[PS10]	Journal article
Schneiderman, Arthur M.	Why balanced scorecards fail	[Sc99]	Journal article
Siegwart, Hans	Kennzahlen für die Unternehmungsführung	[SRS10]	Book
Tuomela, Tero-Seppo	The interplay of different levers of control: a case study of introducing a new performance measurement system	[Tu05]	Journal article

Table 2.4.: Output overview from enterprise management literature

With respect to our research goal (cf. Section 1.1), we consider seven of the identified sources from the field of EA management as particularly important, hence they describe valuables concepts, which we incorporate in the design of our solution. Thus, we provide detailed descriptions of these works in the subsequent subsections. The other works are directly referenced with their contribution in Section 3.

2.3.1. Metrics for Application Landscapes by Lankes

In his PhD thesis [La08], Lankes studies the usage of metrics in the context of application landscape management. For this purpose, he conducts an expert survey (EAMVS) with 22 participants from 19 German organizations and investigates the acceptance, needs, and adoption rate of metrics for the given purpose. According to his results, the majority of the interviewees (52%) confirms the usage of metrics for application landscape management in their enterprises. Additionally, 37% confirm concrete interest in adopting such metrics.

However, 11% state that they are not interested in applying such metrics in their organizations. The most commonly given justifications for this position are:

- The expected efforts for collection and quality management of the required data are considered too high.
- The benefits from the usage of such metrics are either not clear enough or not understood as relevant from businesses point of view and thus, no management support and investment interest for a corresponding measurement initiative are given.
- The missing knowledge regarding the question which metrics are to be used in which situations are considered as problematic.
- Employees resist to the introduction and implementation of metrics because they see a high risk in abusing metrics for observation purposes by their management.

Consequently, Lankes provides an overview of existing metrics for the management of application landscapes based on existing literature and the gathered feedback from the interviewed industry experts. The presented metrics are documented together with corresponding I-Patterns (describing the information demands of each metric with respect to the underlying calculation rule) and are integrated within the *EA Management Pattern Catalog* (EAMPC) [Bu08]. Afterwards, Lankes focuses on the concrete problem of quantifying the failure propagation in application landscapes in depth. For this purpose, he defines and investigates concrete metrics to quantify failure propagation in application landscapes. In addition, he presents simulation techniques targeting the mitigation of failure propagation based on his metrics. For the visualization of his concepts, Lankes applies widely-accepted EA management visualizations, e.g. cluster maps [Wi07b]. Figure 2.9 shows a cluster map used to visualize and

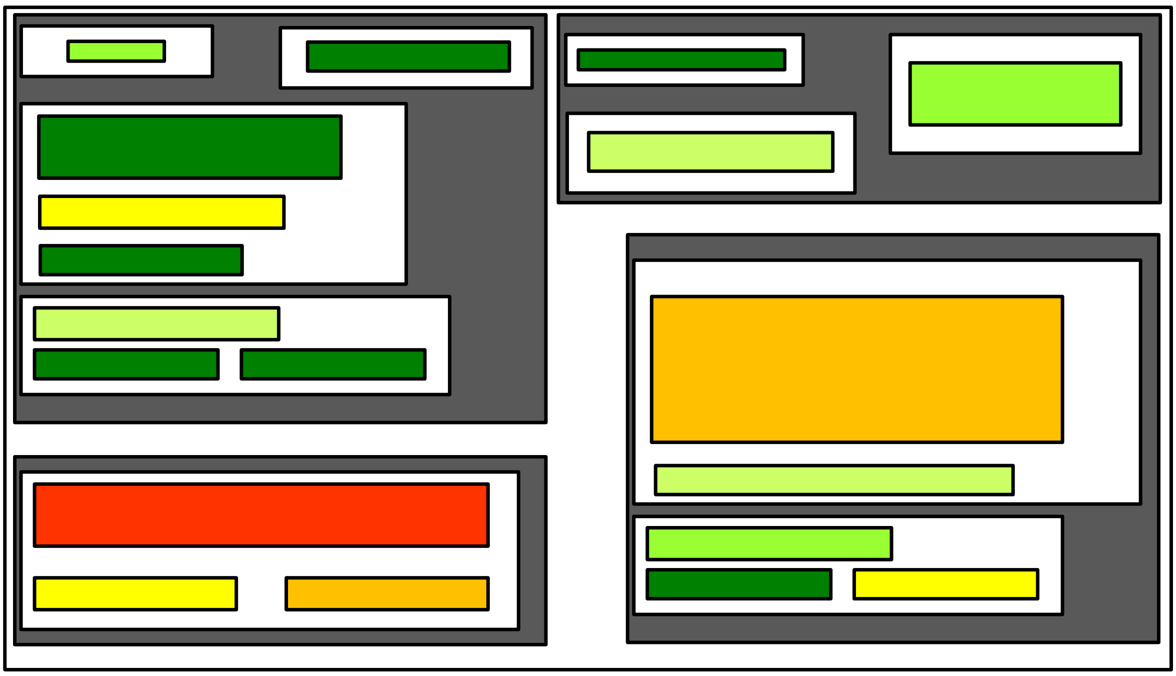


Figure 2.9.: Exemplary visualization of failure propagation metrics according to Lankes [La08]

communicate the results of the application of his metrics to the involved stakeholders. In this visualization, the inner rectangles represent single business applications. The size and color of the rectangles are coupled with Lankes metrics. The color indicates the estimated availability of a given application. Thereby, the higher the estimated availability of an application, the greener the color of the corresponding rectangle. Additionally, the size of the rectangles indicates the extent to which coupled applications are affected in case of failure. For example, big red rectangles indicate that a certain application should be maintained, since it has a low estimated availability (color of the rectangle) and in addition, a failure will affect many other coupled applications (size of the rectangle). This representation of Lankes application landscape management metrics is evaluated as helpful and useful by enterprise architects and managers as described in his thesis. Lankes further presents a prototypical software support for his metrics allowing users to enter their application landscape data and to compute the presented metrics. Additionally, the tooling allows users to perform simulations of concrete countermeasures to mitigate failure propagation.

With respect to our research, we account for the usage of metrics for the quantification of static aspects of a given EA in our work. Further, we incorporate the idea of developing a tool support for the implementation, calculation, and visualization of metrics in our solution.

2.3.2. Assessment of Applications in the Context of Their EA by Addicks

In his PhD thesis, Addicks [Ad10] presents an assessment method (ITEVA) for applications considering their organization-specific EA context. Figure 2.10 illustrates his method, comprising the following three steps:

Figure 2.10.: The ITEVA method for assessment of applications in the context of their EA according to Addicks [Ad10]

1. In the first step *Computation of metrics* (in German “Kennzahlenberechnung”), different properties of the applications are identified, e.g. *availability* of applications. Thereafter, corresponding metrics from scientific sources, e.g. [La08, Kü10] are selected and calculated to quantify specific properties according to the goals and concerns of the involved stakeholders in this process.
2. In the second step of the method, *Classification* (in German “Klassifikation”), concrete assessment criteria for the applications are identified based on interviews with the involved stakeholders from the organization. Thereby, relevant metrics from the first step are selected for each documented assessment criteria. The output of this step is a set of metrics, containing at least one metric for each assessment criteria.
3. In the last step *Aggregation* (in German “Aggregation”), an aggregation of all previously instantiated metrics is performed and the result is a single numerical value, which can be communicated to the management. The aggregation of the metrics thereby takes

different scale types into account and applies *fuzzy-operators* to deal with missing data in the underlying EA model.

Additionally, the author also presents a process model called *EVA*, designed to guide the application of his assessment method. Thereby, concrete roles, responsibilities and artifacts are described in detail to support the organization-specific instantiation of the ITEVA method. In the first step of the EVA method, different assessment criteria are identified based on interviews with the involved stakeholder. Additionally, recommended metrics are selected according to interests of the stakeholders and are initially calculated. In the second step of the method, the metrics are aggregated according to the ITEVA concepts and continuously measured and reported to the involved stakeholders. A parallel step can be executed whenever a new metric is required, an existing one is not relevant anymore for a given stakeholder, or new assessment criteria need to be considered.

The author further presents a software prototype tailored to the support of his ITEVA and EVA methods and designed for the usage by enterprise architects. The tool is evaluated as helpful and useful by industry experts as described in the evaluation chapter of his thesis. The software is written in Java [Go13] and is integrated in the Eclipse framework (Eclipse Rich-Client-Platform) [MLA10]. Users of the tool can enter their application landscape data in the tool and are able to define and manage criteria, metrics, and metrics aggregation according to the ITEVA concepts. Furthermore, the tooling provides visualizations designed to support the communication of the results to involved managers.

With respect to our research, we incorporate the idea of selecting (structural EA) metrics in line with the goals and concerns of the stakeholders involved in an EA metric process in the design of our solution. In analogy to Lankes (cf. Section 2.3.1), the author shows the importance of providing a software support for the management of metrics and its capability of supporting metric visualizations.

2.3.3. Value-oriented EA Management by Stutz

In his PhD thesis, Stutz [St09b] presents a development method for organization-specific EA management metrics based on given business and IT strategies. Thereby, the method is constructed based on the consideration of several existing approaches and techniques for measuring EA benefits. For this purpose, Stutz conducts an extensive literature review and combines the findings in a metrics value based EA management method. Figure 2.11 outlines

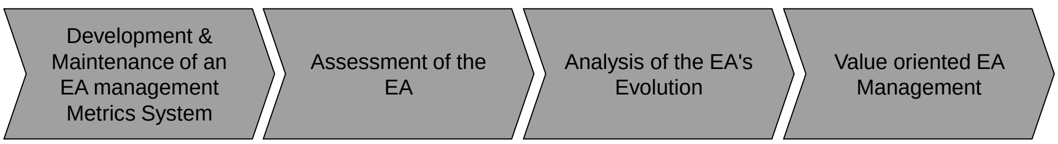


Figure 2.11.: Value oriented EA management method according to Stutz [St09b]

his method. It consist of the following four distinct phases:

1. In the first phase *Development and maintenance of an EA management metric system* (in German “Entwicklung und Wartung des Kennzahlensystems”), an organization-specific

EA management metric system is developed in alignment with the IT and business strategy of the user's organization. Moreover, in this step concrete EA goals and targets as well as planned values for these goals are defined and documented.

2. In the second phase *Assessment of the EA* (in German "Messen der Unternehmensarchitektur"), the current state of the EA is assessed using the EA management metric system developed in the first phase of the method.

3. In the third phase *Analysis of the EA's evolution* (in German "Analyse der Veränderungen der Unternehmensarchitektur"), the changes of the EA with respect to the previously assessed EA state are analyzed.

4. In the fourth phase *Value oriented EA management* (in German "Wertorientierte Steuerung der Unternehmensarchitektur"), concrete countermeasures are defined to ensure the achievement of the predefined EA goals and target/planned values by using the results from the previous step as a management decision support.

One essential aspect of Stutz's approach is the usage of a BSC tailored to the needs of the EA management domain. Thereby, the author highlights the success of the original BSC approach by Norton and Kaplan [KN91] in the field of management controlling and claims, that a corresponding adaptation of the classical BSC perspectives will provide a similar benefits in the domain of EA management. The idea of using a BSC for our domain is also in line with the understanding of other authors, cf. [Sc06, PSP12]. Consequently, Stutz presents the following adaptation of the classical BSC perspectives:

- The financial perspective is named *finances of architecture* (in German "Architekturfinanzen"). This perspective is concerned with the assessment of costs and financial benefits which can be attributed to the EA initiative.

- The customer perspective is named *architecture services* (in German "Architekturservices"). This perspective is concerned with the assessment of the following assessment factors - created EA artifacts and offered EA services, as well as the assessment of the quality of these EA artifacts and services.

- The process perspective is named *architecture processes* (in German "Architekturprozesse"). This perspective is concerned with the assessment of the efficiency and effectiveness of the existing EA processes.

- The human resource perspective is named *architecture assets* (in German "Architekturassets"). This perspective is concerned with the assessment of the following assessment factors - architecture knowledge in the EA team, organization of the EA team, and structure of the EA team.

The adapted BSC plays a key role in Stutz's method. Figure 2.12 outlines the "ideal process" for the application of his method. Thereby, the first activity in his method is the identification and documentation of demands for EA management metrics by the involved management. Thereafter, in the second activity, additional metric demands are identified based on the given IT and business strategy of the organization. Here, the adapted BSC comes into play and is used to develop a balanced metric system concerned with the strategic alignment of the EA management initiative. Additionally, concrete EA management goals are defined and by applying the *Goal-Question-Metric Approach* [BCR94], a corresponding metric system is defined.

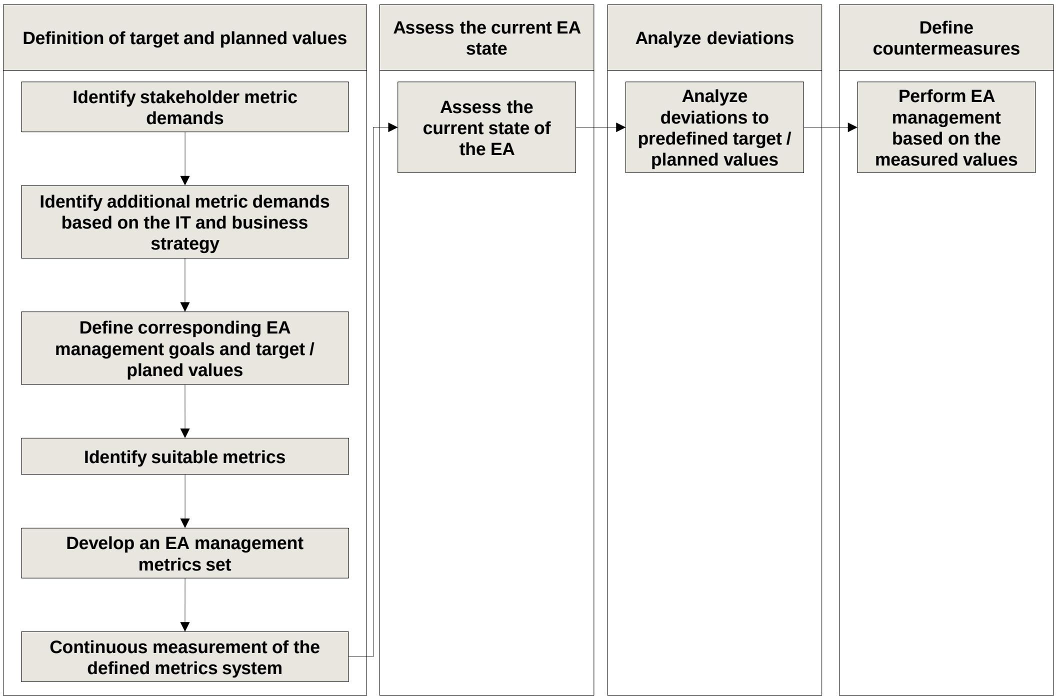


Figure 2.12.: Ideal value based EA management process according to Stutz [St09b]

Afterwards, the metrics are prioritized and aggregated with respect to the expectations of the involved stakeholders. The metric system is documented by a corresponding meta-model proposed by the author, cf. Section 3.4.3. The author further states, that the metric system, which is to be implemented, describes concrete steps which are to be performed to ensure access to the required enterprise data and outlines the benefit of the regular computation of the metric system as management decision support.

With respect to our research, the work of Stutz [St09b] describes several concepts, which we incorporate in our solution. First, metrics are defined based on organization-specific goals in line with the strategy of a given organization and the expectations of the involved stakeholders. Further, the life-cycle management of metrics requires a corresponding metric management method starting from the definition and documentation of metrics. Since EAs reflect organizational changes, a metric management method has to account for the adaptation of existing metrics, as well as for the definition of new metrics and the deletion of metrics no longer needed as a consequence of the changes. Further, the author provides a small number of exemplary metrics developed using his method (in the appendix section of his work). In our understanding, the identification and documentation of practice-proven metrics in the domain plays an important role for our method in terms of creating an organized collection of metric best-practice for specific EA problems. Nevertheless, we decide to not incorporate the idea of using balanced scorecard perspectives in our work. As our present research confirms (cf. [MSM13]), the two aspects *security* and *compliance* have to be considered as additional cross-cutting aspects of the overall enterprise architecture structure (as a consequence of the increasing number of regulatory requirements and laws, in particular from the perspective of big and internationally operating enterprises). However, as also confirmed by Neely [NAK02], the classical BSC fails to account for these aspects, since none of the proposed perspectives accounts for regulatory and security requirements by its definition. In our understanding, the four perspectives presented by Stutz do not account for these aspects as well. Hence, organizations require specific adaptations of the proposed BSC, which however is beyond the scope of our work.

2.3.4. The Enterprise Architecture Value Framework by Plessius et al.

In [PSP12], Plessius et al. present their *Enterprise Architecture Value Framework* (EAVF) as depicted in Figure 2.13. In analogy to Stutz [St09b], the authors support the idea of applying

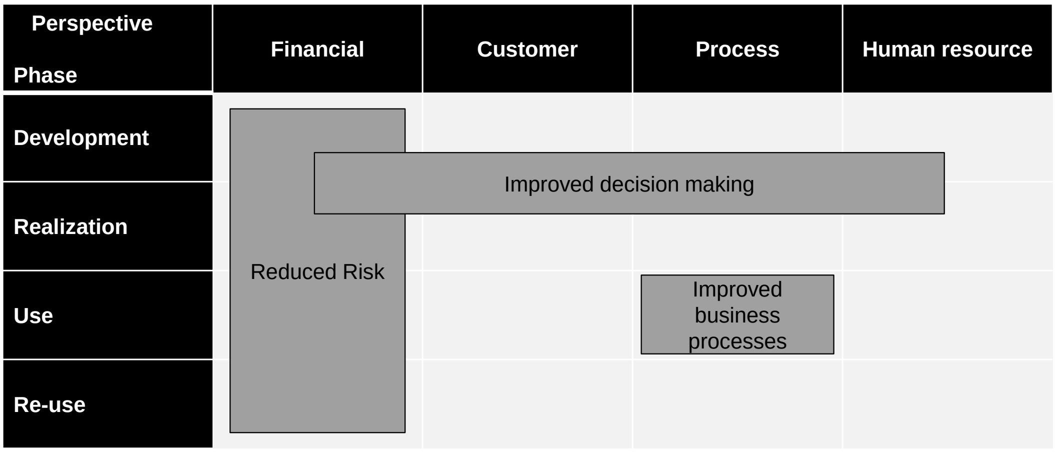


Figure 2.13.: Overview of the EAVF according to Plessius et al. [PSP12]

the BSC concept to the EA management domain. However, they prefer to stick to the original four perspectives and do not present a specific adaptation for the EA management domain. The four classical BSC perspectives are placed on the horizontal axis of their framework. On the vertical axis, the authors place four typical EA management phases, inspired by the *architecture development method* (ADM) of the TOGAF [Th09] framework. These four phases are defined as follows:

- In the *Development* phase, an organization-specific EA is developed and maintained. This phase thereby corresponds to ADM's phases *architecture vision*, *business architecture*, *information systems architecture*, and *technology architecture*.
- In the *Realization* phase, different programs and projects are initialized and executed to implement the changes as specified by the EA documentation. This phase corresponds to ADM's phases *opportunities and solutions*, *migration planning*, and *implementation governance*.
- The *Use* phase starts when the realization phase is over. This phase is characterized by the usage of the introduced changes by the organization. During this phase, data regarding the measurement of benefits and drawbacks of the introduced solutions is collected. This phase corresponds to ADM's phase *architecture change management*.
- The last phase *Re-use* is concerned with the reuse of previously introduced enterprise

architecture artifacts and focuses on the collection of relevant data. This phase corresponds as well to the ADM's phase *architecture change management*.

Thirdly, 12 typical EA management goals are presented based on an extensive literature review - *increased responsiveness and guidance to change*, *improved decision making*, *improved communication & collaboration*, *reduced (IT) costs*, *business-IT alignment*, *improved business processes*, *improved IT systems*, *re-use of resources*, *improve integration*, *reduce risk*, *regulatory compliance*, and *provides stability*. These goals are placed in the EAVF matrix according to the authors' understanding and indicate which goals should be quantified in which EA management phase by which BSC perspective. Further, the authors present a list of six exemplary metrics for the customer column of their framework. Finally, they propose a maturity model for the adoption of their framework in different organizations. This maturity model consists of the four levels - *ad-hoc*, *measurable*, *measured*, and *managed*. The authors state that an application of a metric system for the assessment of benefits related to a given EA management initiative is meaningful first after ensuring that maturity level "measured" is reached by the given organization.

With respect to our research, the authors outline the importance of defining metrics in line with typical EA management goals. Further, the authors emphasize the importance of achieving specific EA management function maturity levels to allow the usage of specific metrics. Additionally, they support the idea of applying a BSC for the management of EA management metrics. However, since Plessius et al. do not propose any adaptation of the classical BSC (in contrast to Stutz), their approach faces the same issues as described in Section 2.3.3. Hence, we decide do not incorporate this idea in our solution.

2.3.5. Predictive, Probabilistic Architecture Modeling Framework by Johnson et al.

In [Jo13a], Johnson et al. present their *predictive, probabilistic architecture modeling framework* (P^2AMF) as depicted in Figure 2.14. This framework is designed to support the analyses

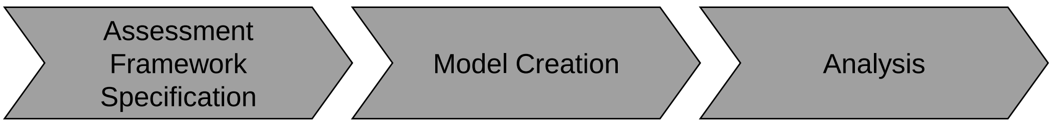


Figure 2.14.: P^2AMF method according to Buschle et al [BJS13]

of specific EA properties, e.g. the availability of an application. In the first step (*Assessment Framework Specification*) of the method, a meta-model for the required analyses is created. The meta-model contains all required concepts, attributes, and relationships between the concepts. Thereby, the meta-model not only describes the allowed content, but also documents how concrete characteristics of the concepts influence each other. In the second step (*Model Creation*) of the method, scenarios of interest from the perspective of the involved enterprise architects and managers are identified and documented. For each scenario, an instantiation of the meta-model is performed, containing all relevant instance data to justify the decision regarding the correctness and usefulness of the model. In the final third step (*Analysis*) of the method, the analyses are performed and quantitative values are computed. Furthermore, corresponding visualizations of the results are created to communicate the results to involved stakeholders (cf. [BJS13]).

The P^2AM framework [Jo13a] is based on a combination of OMG's *unified modeling language* (UML) [Ob11b] and *object constraint language* (OCL) [Ob12] which allow architects do model a (software) system by using UML and to constrain its structure and behavior by using OCL statements. UML is frequently applied in the field of EA management for modeling purposes, however, the authors claim that the support offered by the combination of UML and OCL does not fully satisfy all requirements for the analysis of EA properties. The authors show, that enterprise architects always have to consider both—the EA meta-model and its instance model (class diagram and object diagram in terms of UML) as a base for architecture analyses. In addition, the authors highlight the need for considering two types of uncertainty on the instance level, which are typical for the field of EA management. First, certain instance data can be missing. Second, existing data can be outdated. To address these two issues, the authors propose to model uncertainty and existence to each attribute, concept and relationship of the model. They further state that attributes may be even stochastic, therefore, the initial values of the attributes should be expressed as a distribution of probabilities.

Consequently, the authors propose an extension of the OCL language by the introduction of two attributes named *existence* (one for class properties and one for relationships) and the specification of attribute values by means of probability distributions. Based on these changes, the authors show that their solution enables proper probabilistic inference over OCL expressions. Furthermore, the authors provide a tool support (as a Java-based extension of the *Eclipse Modeling Framework* (EMF) [St09a]) described by Buschle et al. in [BJS13]. This tool allows the integration of both—the given EA meta-model and the given EA instance model. Further, it supports the definition of prediction models and thus supports EA scenario building and EA analyses. The software prototype supports different visualizations of the results for the purpose of communication to the involved stakeholders.

With respect to our research, the authors outline the importance of measuring behavioral aspects of the EA system. For this purpose, the authors highlight the importance of the link between the underlying qualitative EA model and quantitative models concerned with the measurement of behavioral aspects. Further, a query language is required to enable a model-based computation of quantitative models. Additionally, the authors show the importance of providing a tool for the implementation of quantitative models. Nevertheless, we decide to not incorporate the idea of modeling uncertainty in our solution. In our understanding, this aspect should be taken into account by organizations first when a certain maturity level of a given EA metric initiative is reached. Based on our experience from several talks with experts from industry, organizations starting with the application of quantitative models in our domain face several data and empowerment challenges during the selection, definition, and implementation of metrics. Hence, we consider the modeling of missing or outdated EA model elements (and data) as too advanced and beyond the scope of this thesis.

2.3.6. Enterprise Architecture Benefits by Niemi

In [Ni08], Niemi presents the results of an extensive literature review concerned with typical benefits associated with EA management. Next to several concrete benefits found in literature, the author additionally presents a classification schema for these benefits regarding following two aspects as sketched in Figure 2.15:

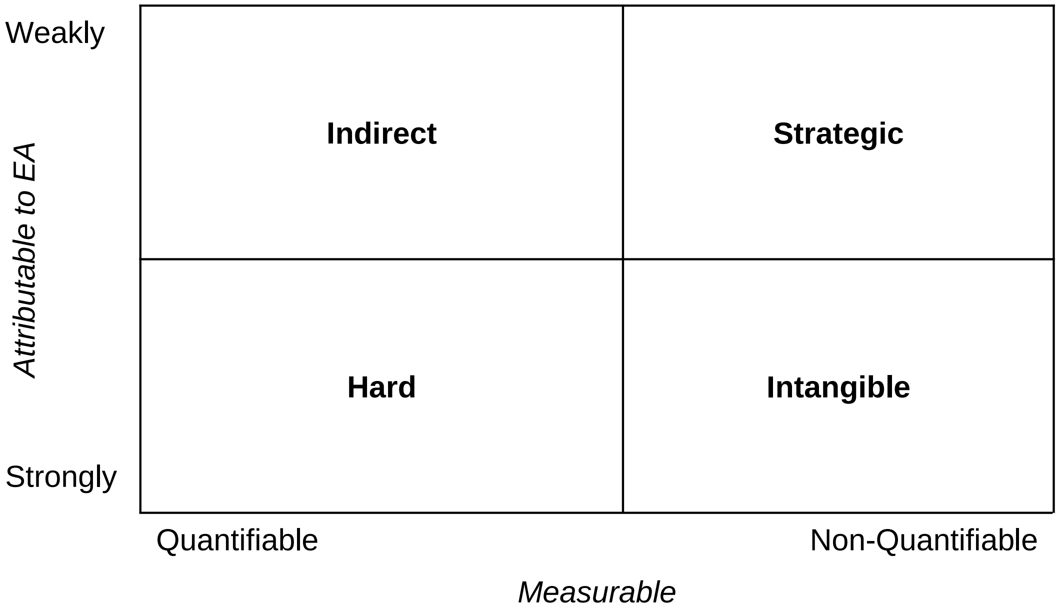


Figure 2.15.: EA benefits categorization according to Niemi [Ni08]

1. The vertical axis of the presented categorization describes to which benefits can be attributed to an EA management function.
2. The horizontal axis of the categorization describes the extent to which benefits can be objectively quantified.

The author further defines the following four distinct subcategories:

- *Hard benefits* can be objectively quantified (e.g. in monetary or other numerical values) and in addition can be completely attributed to a given EA. Moreover, sorted with respect to the citation frequency of the benefits in literature, the author provides the following six examples for hard benefits - *increased economies of scale*, *increased interoperability and integration*, *increased reusability*, *increased standardization*, *reduced cost*, and *shortened cycle times.*
- *Intangible benefits* cannot be quantified objectively, but can be easily attributed to a given EA. According to Niemi, the following three benefits for this subcategory are the most frequently cited ones in literature - *evolutionary EA development & governance*, *improved decision making*, and *provides a holistic view of the enterprise.*
- *Indirect benefits* can be objectively quantified, but cannot be easily attributed to a given EA. Based on his results, the author provides following twelve frequently cited indirect benefits - *improved alignment with partners*, *improved asset management*, *improved business processes*, *improved customer orientation*, *improved innovation*, *improved management of IT investments*, *improved risk management*, *improved staff management*, *increased efficiency*, *increased market value*, *increased quality*, and *reduced complexity.*

- *Strategic benefits* cannot be quantified objectively in general and in addition cannot be completely attributed to a given EA. According to Niemi, "strategic benefits are positive effects that are realized in the long run and are typically affected by a multitude of factors". The author further provides the following six most frequently cited strategic benefits in literature - *improved alignment to business strategy*, *improved business-IT alignment*, *improved change management*, *improved communication*, *improved strategic agility*, and *increased stability*.

According to Niemi's results, the majority of the typical benefits associated with EA management either cannot be objectively quantified, or cannot be completely attributed to a concrete EA (or even both). These findings are in line with our observations regarding the usage of concrete EA management metrics by German organizations as documented in our EA management metric catalog and described in Chapter 3.5.

With respect to our research, Niemi [Ni08] emphasizes the importance of linking quantitative models for the measurement of benefits to the underlying EA. Further, the author shows that not all of the benefits associated with EA management can be objectively quantified, e.g. *business IT alignment*.

2.3.7. A DSL-based Method for Performance Measurement by Strecker et al.

In [St12], Strecker et al. present their *MetricM* approach - a domain-specific modeling method to support the design and use of performance measurement systems. For this purpose, the authors first perform an extensive literature review in the domain of enterprise performance measurement and derive the following list of requirements their MetricM *domain-specific modeling language* (DSL) has to fulfill:

1. *Rational* - according to the authors, a DSL based method should provide concepts that allow for a differentiated representation of the rationale behind a metric and metric relationships. Additionally, the method should support the justification of the existence of metrics and of their relationships to each other. Consequently, the DSL must provide corresponding concepts for a general description of metrics and must support the management of metric relationships, e.g. which metrics are used by a given metric.
2. *Coherence & consistency* - the method must support the design of a coherent and consistent metric system. Consequently, the DSL must provide explicit conceptualization for the relationships management between metrics.
3. *Multiple perspectives and levels of abstraction* - the method must provide meaningful representation of the metric system on various levels of abstraction tailored to the needs of different stakeholders involved. Consequently, the DSL must support different views and representation notations of the metric system.
4. *Organizational context* - the method must account for specific details in a given organizational context. Consequently, the DSL must support the adaptation of the metric system to a given organizational context, e.g. specific process names, actors and responsibilities.
5. *Organizational goal* - the method must support links between the metric system and the organizational goals. Therefore, the DSL must provide a conceptualization of these

links and must provide concepts to account for organization-specific goal properties, e.g. predefined planned or target values.

The authors present a corresponding meta-model (*MetricML*) (cf. Figure 2.16) supporting all required conceptualizations, and describe a DSL called *MetricM* supporting the all of the presented requirements. For more details regarding the defined DSL semantic, syntax, and notation of MetricM, we refer to the original source [St12]. With respect to our research,

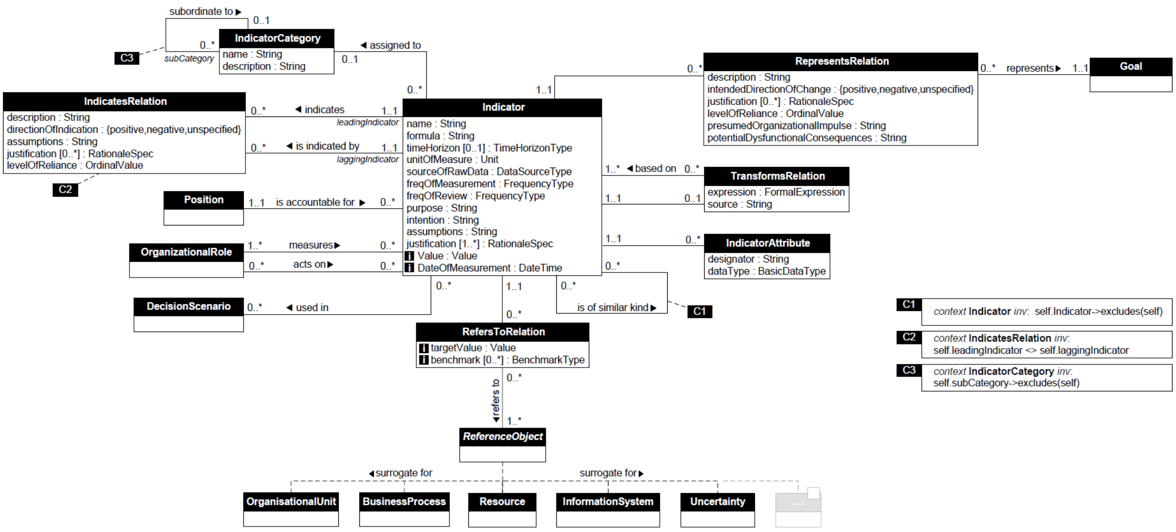


Figure 2.16.: The MetricML meta-model according to Strecker et al. [St12, p. 18]

we incorporate the idea of defining a DSL for the definition and management of queries to support the calculation of quantification models in our work. Further, we consider the need for a comprehensive metric description structure as an indispensable part of a holistic management process. In analogy to Lankes [La08] and Addicks [Ad10], the authors emphasize the importance of links between goals and quantitative models.

2.3.8. Summary

The presented approaches from the EA management domain provide several valuable foundations, which we correspondingly incorporate in our work as illustrated in Figure 2.17. Nevertheless, based on their different research interests and specific scopes, none of the presented approaches meets our research goal. In particular, none approach presents an organized collection of practice-proven metrics for the EA management domain with corresponding links to underlying general EA management goals and concerns. Further, none of the approaches presents a comprehensive metric description structure, comprising only indispensable metric description elements to support the organization-specific definition, instantiation and implementation of metrics. In addition, neither Stutz's [St09b] nor Plessius's [PSP12] metric life-cycle management method considers the usage of metric best-practices and even more, none of them provides a corresponding software support accounting for the implementation of metrics. Additionally, none of the presented software solutions is integrated within a commercial EA management tool. In addition, the presented solutions do not explicitly focus on collaborative

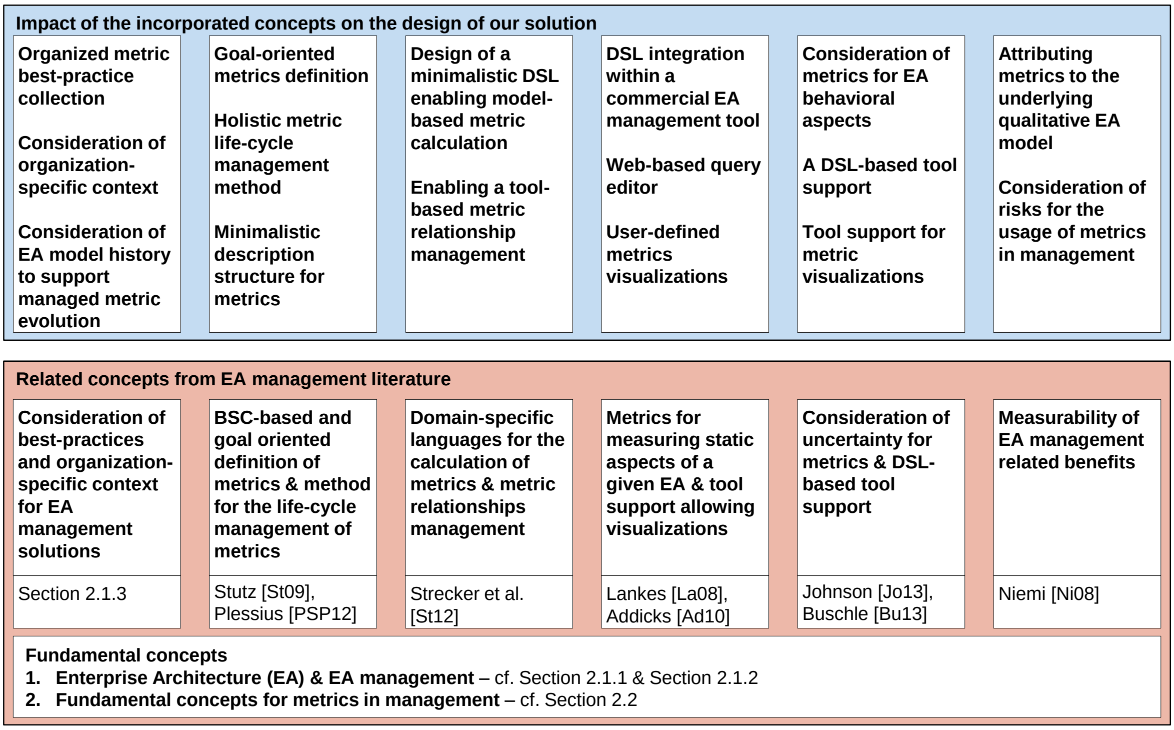


Figure 2.17.: Incorporated concepts from related works in our solution

management of both—qualitative EA models and related quantitative models by their design. However, in our understanding, a tool, designed to allow a holistic metric life-cycle management, has to empower collaboration and to support a lean and emergent EA management approach. Based on the results from our literature review, we can confirm that our research incorporates several concepts from the exiting literature on the one hand, and provides a new contribution to the body of knowledge in this field on the other hand.

CHAPTER 3

Concept Development

In this Chapter, we present the fundamental concepts developed during our research. Firstly, we present identified usage scenarios for quantitative models (metrics) in the EA management domain by taking a systemic perspective on an enterprise (Section 3.1). Thereafter, we discuss the existing terminology regarding the usage of quantitative models in management and introduce the terminology for our research (Section 3.2). Then, we present typical risks, problems and suggested countermeasures associated with the usage of metrics in management. Additionally, we present a prioritization and adaptation of these risks and countermeasures for the domain of EA management (Section 3.3). Afterwards, we propose a generic description structure for the documentation and management of EA management metrics (Section 3.4). Then, we introduce a collection of EA management metrics observed in German industry and uniformly described by the application of our description structure (Section 3.5). Lastly, we present a holistic life-cycle management method for EA management metrics integrated within the BEAMS framework (Section 3.6).

3.1. Identified Usage Scenarios

Before starting the discussion and presentation of concrete usage scenarios for metrics (quantitative models) in the EA management domain, we highlight the importance of *systemic thinking* for our research. The term systemic thinking was coined by Jay Wright Forrester [Fo94] and describes the modeling and analytical thinking abilities required for the application of his *system dynamics* approach. This approach supports the understanding of the behavior of complex systems over time [Fo68, Fo71, Fo94] and has gained significant attention in enterprise management over the years as the high number of citations and publications in the field confirms. The system dynamics approach deals with feedback loops and time delays that affect the behavior of a system. The strength of this approach is that "it can accept the complexity, nonlinearity, and feedback loop structures that are inherent in social and physical systems" [Fo94]. According to [St00], in system dynamics, a problem or a system is represented as a causal loop diagram. A causal loop diagram is "a simple map of a system with all its constituent components and their interactions. By capturing interactions and consequently the feedback loops (cf. Figure 3.1), a causal loop diagram reveals the structure of a system. By understanding the structure of a system, it becomes possible to ascertain a system's behavior over a certain time period" [Me08].

Figure 3.1 provides an example of a casual loop diagram [St01]. The causal loop diagram

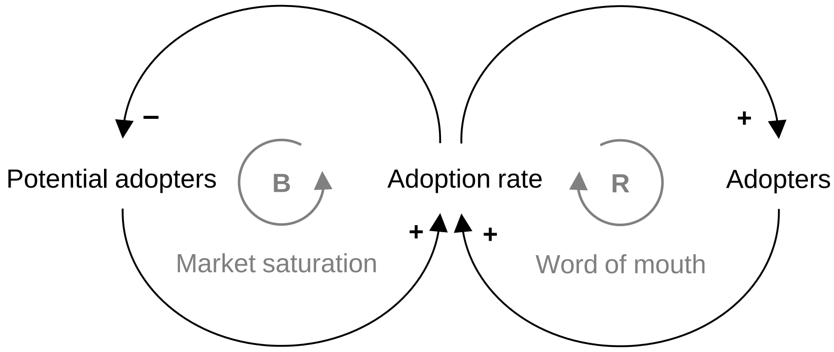


Figure 3.1.: Casual loop diagram for the adoption rate of a new product based on Sterman [St01]

represents the system behavior (adoption rate) for the market introduction of a new product. The diagram consists of two feedback loops which interact [Fo68] as follows:

- The positive reinforcement (labeled R) loop on the right side of the diagram indicates that the more persons have adopted the new product, the stronger is the impact of the word-of-mouth effect towards the product adoption. Put in other words, the number of positive references for the product (e.g. demonstrations, reviews and recommendations) will increase over time. From an enterprise point of view, one can expect, that this positive feedback will generate continuously growing sales.
- The second feedback loop, on the left side of the diagram, represents the negative reinforcement (labeled B). This feedback loop supports following hypothesis - the number of sales cannot continuously grow forever, since the number of potential adopters is limited and is decreasing over time. From an enterprise point of view, one can expect, that the ratio of sales will decrease rapidly as soon as the market is saturated.

To perform more advanced quantitative analysis, causal loop diagram are transformed to so-called *stock and flow diagrams* [Ro82]. Thereby, a *stock* is the term for any entity that accumulates or depletes over time, whereas a *flow* is the rate of change in a stock. For the example above, Figure 3.2 shows a possible transformation of the casual loop diagram into a stock and flow diagram [St01]. In addition to the casual loop diagram, in this model, additionally so-called *innovators* (early adopters), imitators (adopters as effect from the word of mouth influence) and the probability that a potential adopter has not yet adopted the product at a given point in time are modeled.

The stock and flow model consists of two stocks and one flow as follows:

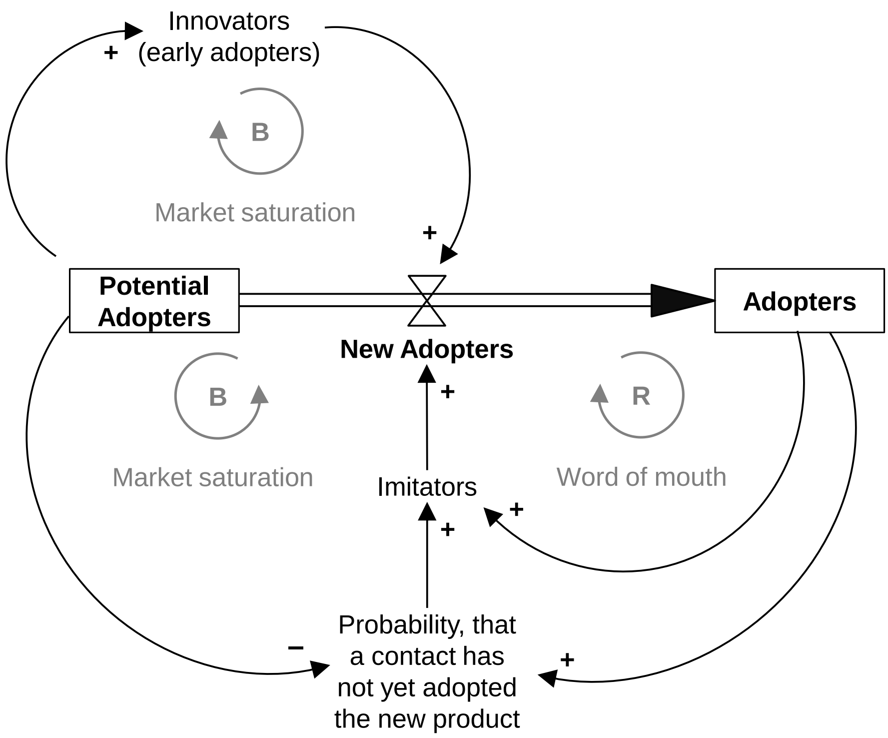


Figure 3.2.: Stock and flow diagram for the adoption rate of a new product based on [St01]

- the potential adopters are transformed to the stock *Potential Adopters*,
- the adopters are transformed to the stock *Adopters*, and
- the flow *New Adopters* is defined. Thereby, for every new adopter, the Potential Adopters stock is decreased by one and in parallel, the Adopters stock is increased by one.

For more information regarding system dynamics we refer to Forrester [Fo68, Fo71, Fo90, Fo94] and for additional examples and the equations used for the above two examples we refer to Sterman [St01].

As Forrester describes in [Fo94], people frequently struggle with the proper application of his approach. To address this issue and to improve this situation, he presents in [Fo94] a detailed process description for the application of the system dynamics approach as illustrated in Figure 3.3. The process consists of the following six steps as described above:

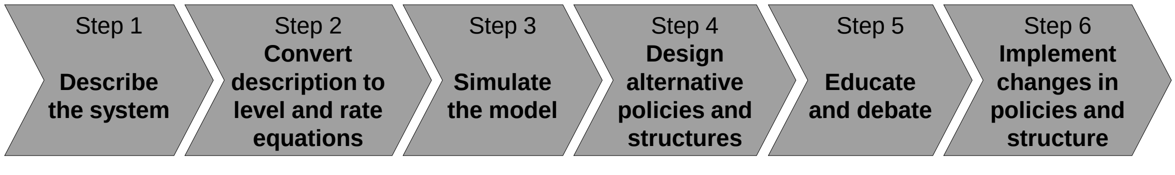


Figure 3.3.: The system dynamics process according to Forrester [Fo94]

1. The first step (*describe the system*) is performed when an "[..] undesirable system be-

havior that is to be understood and corrected". Thereby, a system description and a hypothesis targeting the understanding of the observed negative system behavior are documented.

2. In the second step (*convert description to level and rate equations*), a formulation of a simulation model accounting for the hypothesis defined in the previous step is created. Thereby, by writing equations a modeler can identify inconsistencies and can reveal gaps in the simulation model compared to the modeled reality and then both—the predefined hypotheses and the current simulation model have to be revised. This step is iterated until the modeler cannot identify any inconsistencies to reality.
3. In the next step of the process (*simulate the model*), simulations based on the developed model are performed. If the simulation leads to an unrealistic behavior, the model and the hypotheses are to be revised again. As Forrester states, "[..] there is no way to prove validity of a theory that purports to represent behavior in the real world. One can achieve only a degree of confidence in a model that is a compromise between adequacy and the time and cost of further improvement."
4. In the next step (*design alternative policies and structures*), alternative policies towards the improvement of the systems behavior are developed and tested. The alternative can originate from observations during the first three process phases, from the experience of the analyst or from related literature sources.
5. The fifth step of the process (*educate and debate*) tackles the implementation of the identified improvements. Thereby, it is essential to ensure proper management support and empowerment to implement the desired system structure change. Thus, the analyst requires good skills in communication of the results, consultancy and education of involved stakeholders to ensure the required management support.
6. In the last step of the process (*implement changes in policies and structure*), the identified changes are implemented, and continuous monitoring of the results is performed.

Equipped with this knowledge, in this thesis, we take a systemic view on an enterprise as depicted in Figure 3.4 and previously published by us [MRM13].

According to Berg-Cross [BC07], an enterprise is a system consisting of its structure and its behavior. The system structure is the holistic composition of its elements and the relationships between these elements. Therefore, we use the EA as the model representing the system structure. The (enterprise) behavior refers to the different (enterprise) variables, their functions and relationships. A system's behavior is constrained by its structure [Fo68]. However, understanding the relationship between a system's structure and its behavior is very difficult [Gr73]. As described in Section 2.1.3, enterprises face many different changes over time, which makes it even harder to relate the system's structure to its behavior [BC07].

As a consequence, to predict an enterprise's behavior over time, enterprise architects have to understand both—the statics of the system's structure (EA) and the dynamics of the system's behavior to adequately ensure architectural alignment, proper integration and agility of the EA management function. This can be achieved by applying the system dynamics approach by Forrester [Fo94] and by adopting his systemic thinking concept. As an important consequence, enterprise architects can introduce **direct** changes only to the EA based on predefined hypotheses regarding the timely delayed (enterprise) system behavior. Since the

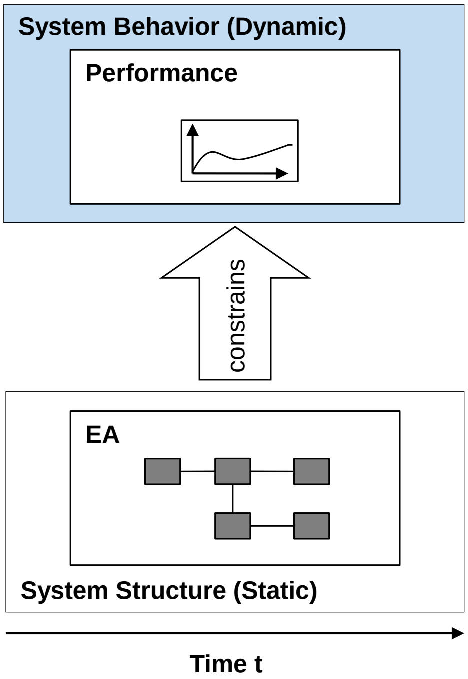


Figure 3.4.: A systemic view on an enterprise [MRM13]

EA constrains the possible behavior of the (enterprise) system [BC07], the expected changes in the system behavior are done only **indirect** and are timely delayed. Thus, enterprise architects have to follow the dynamic systems process of Forrester [Fo94] and in particular to develop excellence in the step five (educate and debate). This is important, since in this steps the EA management stakeholders are informed, consulted and educated about different change alternatives and a common understanding of the solution, cost, duration as well as expected effects are established. Hereby, the focus must be set on the provision of transparency regarding:

1. the costs, effects and duration of the architectural change and
2. the duration until interrelated improvements in the system's behavior can be expected.

In our model, it is important to understand the different time aspects of the structural and behavioral layer. On the one hand, we consider changes in the structural layer as time-discrete events, i.e., every system model state (EA state) corresponds to exactly one concrete point in time and all structural changes are attributed to exactly one EA state. Additionally, there is always only one **current EA state**, whereas several planned, target and past EA states can exist. On the other hand, we consider changes in the system's behavior (enterprise system) layer as timely-continuous (i.e., change events, stock and flows). For example, the failures of a given application should be understood as change events on the system's behavior layer. In analogy, the costs of an application (i.e., the cost aggregation of e.g. maintenance cost, development cost, and license cost) should be understood as a flow. Therefore, both system behavior properties - the change events of failures and the costs flow of applications should be investigated for a given period of time and not for a given point in time.

Starting with this systemic view on an enterprise and accounting for the presented relevant EA management approaches in Chapter 2, we identified four typical usage scenarios for metrics (quantitative models) in the domain of EA management as depicted in Figure 3.5

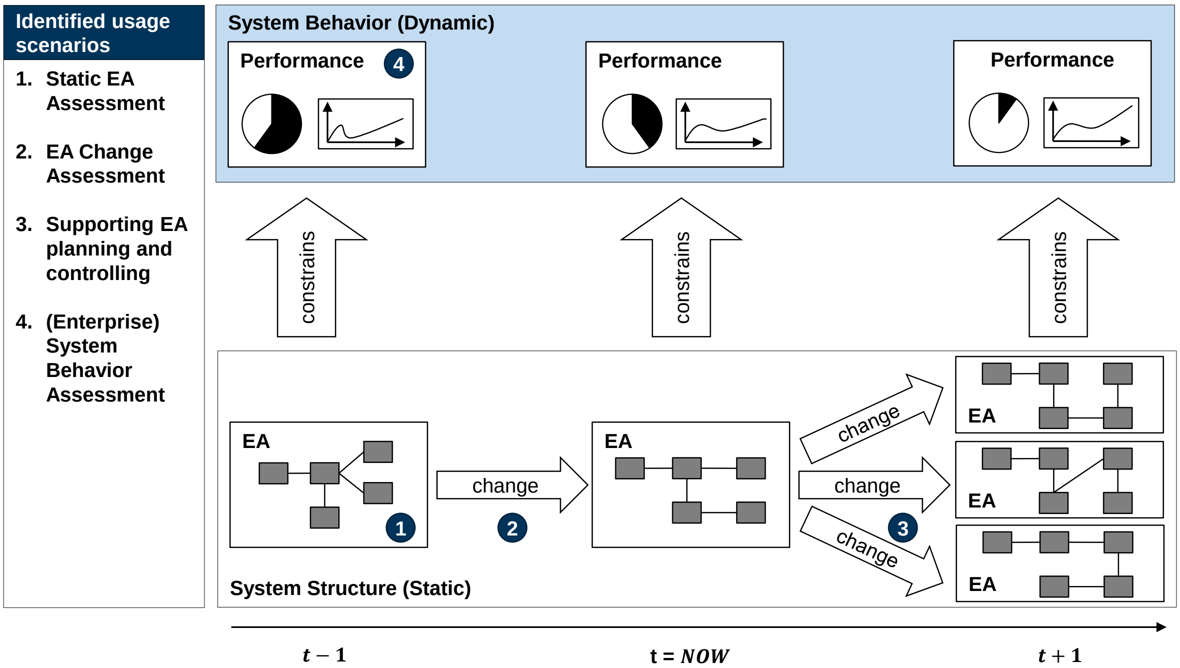


Figure 3.5.: Usage scenarios for quantitative models (metrics)

Firstly, all of the presented related approaches describe metrics used to quantify different (static) aspects of the underlying EA (structure of the system), e.g. "the number of coupled applications which are affected by a failure of given application" [La08]. In this usage scenario, the presented metrics are used to quantify EA model elements, i.e., concepts, attributes and relationships between the concepts. Thereby, in the terminology of UML [Ob11b], metrics are used to quantify both—class as well as object models [Jo13a]. In addition, these metrics should be applicable to any given (snapshot of the) EA state [St12]. We name this usage scenario **static EA assessment**.

Secondly, as described by EA literature [Sc06, St09b, St12, PSP12], enterprise architects must ensure the achievement of predefined planned and target values for the intended EA change (in terms of a continuous EA change controlling and monitoring). Since metrics for the static EA assessment are applicable to any given EA state, metrics can be used to quantitatively compare different EA states and thus to quantify the EA change as well as the corresponding EA change processes (EA projects). For instance, the above example metric can be applied on snapshots of the EA representing different quarters of a year. Based on the comparison of the metric values, enterprise architects can analyze how the number of affected systems by a given application has changed over the time and, if an unexpected negative development is identified, they can start to investigate the causes of this development (e.g. by reviewing and investigating the project portfolio, or by interviewing the responsible application owner regarding unexpected or not documented changes in the deployment model of the given application). The information collected in this case can be used as input for future improvement initiatives or the definition of architectural guidelines. We name this usage scenario **EA change assessment**.

As described by Lankes [La08] and Johnson [Jo13a], metrics (quantitative) models are used to support scenario building and for simulating future EA states, e.g. - changing the coupling topology between applications to reduce the impact of failure propagation. For this purpose, different future EA states (scenarios) can be defined and assessed using metrics for static EA assessment. In this way, two benefits can be achieved. On the one hand, enterprise architects can provide quantitative data to their management as justification for improvement investment requests. On the other hand, enterprise architects can define a concrete road map of the required architectural changes to implement a given scenario by performing a *gap analysis* [Bu11] between the current and the desired future EA state. Afterwards, the road map can be translated to concrete EA change projects and the desired architectural change can be monitored and controlled by applying metrics from the EA change assessment usage scenario. We name this usage scenario **Supporting EA planning and controlling**.

Finally, metrics are used to assess (simulate) the dynamic behavior of the (enterprise) system [La08, Ad10, St12, Jo13a]. In this usage scenario, the system's behavior is assessed in terms of improvement of different performance aspects, e.g. the failure tolerance or costs of applications for a given time period. Thereby, based on simulated changes in the EA, the behavior of the system is quantified to justify the correctness of the changes or to identify and investigate unexpected negative changes in the behavior. We name this usage scenario **(Enterprise) System Behavior Assessment** (cf. Figure 3.5).

3.2. Terminology

Our research question 2 targets the existence of a comprehensive terminology for the usage of quantitative models in the reviewed management literature and is important with respect to the definition of a terminology base for this thesis. However, while processing the identified sources (cf. Section 2.3), we realized that the authors use very heterogeneous terms in their works. Thus, from a reader's perspective, we experienced irritation while studying the sources. For example, we discovered the following terms used in the reviewed literature: *metric*, *performance metric*, *measurement*, *performance measurement*, *measure*, *performance measure*, *indicator*, *key indicator*, *key performance indicator*, *leading indicator*, *lagging indicator*, *result indicator*, *key result indicator* and *critical success factor*.

To make a first step towards the definition of the terminology of our work, we decided to investigate the most frequently used terms in the identified English literature in more depth. In her Master's thesis, which we supervised, Subashi [Su13] showed, that the terms *indicator*, *key performance indicator* (KPI), *measurement* and *metric* are the most frequently used ones. Table 3.1 shows which authors use which of these four terms (by providing a concrete definition) according to [Su13].

Please note, that the list presented here was update in March 2014, and now contains four additional sources that are not included in Subashi's thesis. The reasons for this are:

1. We performed the *Input* step of our literature review one more time in March 2014, to ensure that we do not miss relevant publications after the time when Subashi performed

Citation key	Indicator	KPI	Measurement	Metric
[AGW11]				
[Bu09]		✓		✓
[BJS13]				
[Jo13a]				
[KAV05]				✓
[La08]				✓
[Ni08]				✓
[Th09]		✓		
[PSP12]	✓			
[St12]	✓			
[Wi07a]	✓			✓
[BCR94]				✓
[FGM07]	✓	✓		✓
[St06]		✓		✓
[Bi05]	✓		✓	
[Bo03]	✓		✓	✓
[BP00]				
[Ec10]		✓	✓	✓
[Ec90]				
[HK98]	✓			
[KN91]	✓			
[Ko95]				
[LDH08]	✓	✓	✓	✓
[LE02]	✓			
[NAK02]			✓	✓
[Ne02]			✓	✓
[Ne99]			✓	✓
[NGP95]			✓	✓
[Pa10]	✓	✓	✓	
[Pe98]	✓		✓	
[PS10]	✓			
[Sc99]	✓		✓	✓
[Tu05]	✓		✓	✓

Table 3.1.: Overview of the most frequently used terms in the related sources

her research (to be precise, her review was performed between September 2012 and January 2013 and her *keyword search* was performed in September 2012). While we were redoing the *forward authors search*, we discovered the two sources [Jo13a, BJS13] in the publications list of Prof. Pontus Johnson, which were published after Subashi's review had been performed. The same applies to the source [PSP12], which was published end of October 2012 in the proceedings of the TEAR 2012 conference.

2. We found the work of Niemi [Ni08] by incident, while working on another topic, however, the article does not have keywords and thus was not found by our search. Since we see the

presented EA benefits classification framework in this work as relevant for our research, we include this publication also to the list of related work.

Additionally, the review in [Su13] showed that these commonly accepted terms are used in homonymous manner. The following excerpt of the reviewed sources shows the homonymous usage of the terms *performance indicator* and *key performance indicator*.

According to Lebas and Euske [LE02] "[..] 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". Popova [PS10] defines performance indicators as "a quantitative or qualitative indicator that reflects the state/progress of the company, unit or individual". Parmenter [Pa10] does not provide an explicit definition of the term, however he states that "performance indicators tell you what to do".

Eckerson [Ec10] provide the following definition of a KPI: "A KPI is a metric measuring how well the organization or individual performs an operational, tactical, or strategic activity that is critical for the current and future success of the organization. [..] A metric is the standard measurement of a known object or activity. [..] The measurement is the result or output of measuring an object or activity." Popova [PS10] defines a KPI as a "[..] subset of indicators, that can give a representative picture of the performance and the costs of measuring and monitoring are reasonable." According to Franceschini [FGM07], KPIs are "indicators that "properly" represent a process", where process is described as "an integrated system of activities that uses resources to transform inputs into outputs". Parmenter [Pa10] defines KPIs as: "KPIs represent a set of measures focusing on those aspects of organizational performance that are the most critical for the current and future success of the organization. KPIs tell you what to do to increase performance dramatically". According to Steinberg [St06], "KPIs are metrics that are used to indicate the performance level of an operation or process. They are used to provide a basis for actionable management decisions". Fitz-Gibbon defines a KPI as "[..] an item of information collected at regular intervals to track the performance of a system [enterprise]".

As the previous examples show, a comprehensive terminology regarding the usage of quantitative models in management literature is missing. Thus, we are forced to provide our own definitions for this thesis. Hence, we focus on the terminology required for quantifying the systems's structure (EA). Thereby, we observe enterprises by considering the available EA model data. In this context, we distinguish between quantitative models concerned with the assessment of time-discrete snapshots of the enterprise structure (EA) and quantitative models concerned with the assessment of the time-continuous behavior of the EA, by considering time-series of corresponding changes in the EA. The EA in turn, is a collection of EA models (conceptual and instance models) with the goal to provide an adequate representation of the reality for a given purpose and a given stakeholder (abstraction by modeling) [Sc11, Bu11, Ke07, Sc09, Gr12]. According to Buckl [Bu11] the EA has different states, whereby always one current state should be defined, and a target as well as several planned EA states might be defined. Additionally, in the domain of EA management, many different modeling languages are introduced and applied. In this thesis, we stick to *UML* [Ob11b] and thus, we use UML's semantic, syntax, and notation for our models. In our understanding, an EA model consists of *EA model elements*. According to the UML specification, model elements can be either classes, class attributes or relationships between classes. EA models (and their EA model elements) provide the foundations for qualitative descriptions of the system's structure (EA) [Wi07b, Er10, Bu11, Sc11]. To empower the capability of quantitative descriptions of the system's structure (EA), we define the term **EA metric** as follows (cf. Figure 3.6):

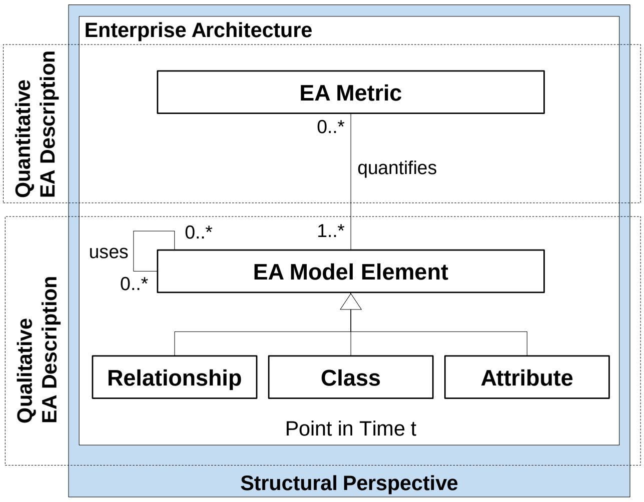


Figure 3.6.: Conceptualization of an EA metric

Definition: EA metric
An EA metric provides quantitative information regarding the system's structure.

Additionally, metrics are collected at regular intervals. An EA metric is applicable on different EA states and quantifies exactly one EA model, i.e., at least one EA model element for a given point in time (time-discrete). Equipped with this definition, EA metrics provide the required support for the previously presented three usage scenarios for quantitative models in the EA management domain – *static EA assessment*, *EA change assessment*, and *supporting EA planning* as the subsequent example shows.

Example 3.1: Example for the usage of an EA metric. As we observed during a cooperation with a German financial service provider, a regulation authority requested the following structural assessment (compliance requirement) from the organization – with respect to predefined criteria by the authority, the enterprise has to deliver a classification of business processes in *critical* and *not critical*. Additionally, the enterprise has to ensure, that the *business applications* providing the IT support of the critical business processes are classified as *critical* as well. Hence, every *critical business application* must have an assigned *IT continuity plan*. An IT continuity plan describes strategies, which are implemented in the case of application failure (i.e., rerouting requests to other data centers). To address this issue, the organization decided to involve the EA management team in the implementation process of this requirement and requested the definition and monitoring of a corresponding EA metric. Consequently, an EA metric with the description "measurement of the coverage of IT continuity plans in respect to critical business processes" was defined. The calculation rule of this EA metric was "number of business critical processes relying on critical business applications covered by an IT continuity plan, divided by total number of business-critical processes". Figure 3.7 illustrates the minimal EA model (in terms of minimal number of EA model elements) required for the computation of this EA metric according to our metric catalog [Ma12a] (cf. Section 3.5). The target value for this EA metric is set to 100%, since high financial fees are expected in the case that the requirement is not fulfilled

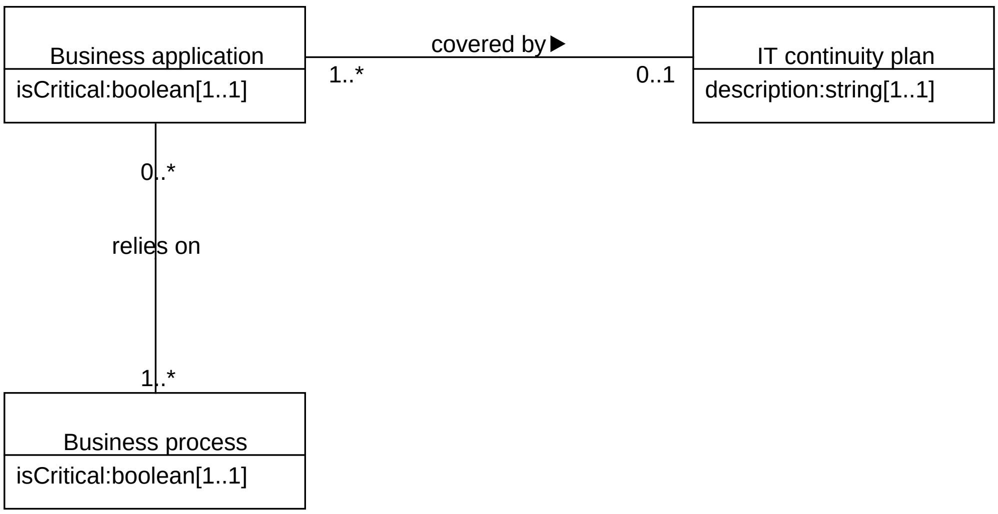


Figure 3.7.: Minimal EA model for the example EA metric [Ma12a]

Secondly, we require a specific type of quantitative models to assess the (enterprise) system's behavior over specific time intervals. Thereby, every EA model element has a behavior, e.g. a business application is available to a specific degree for a given period of time. Hence, behavior aspects require the existence of corresponding EA structure elements and the consideration of change events and flows as depicted in Figure 3.8. Further, by applying causal loop as well as stock and flow diagrams, enterprise architects can create simulation models for the EA towards specific improvements, which however is beyond the scope of this thesis. Therefore, we define the term **performance indicators** as follows:

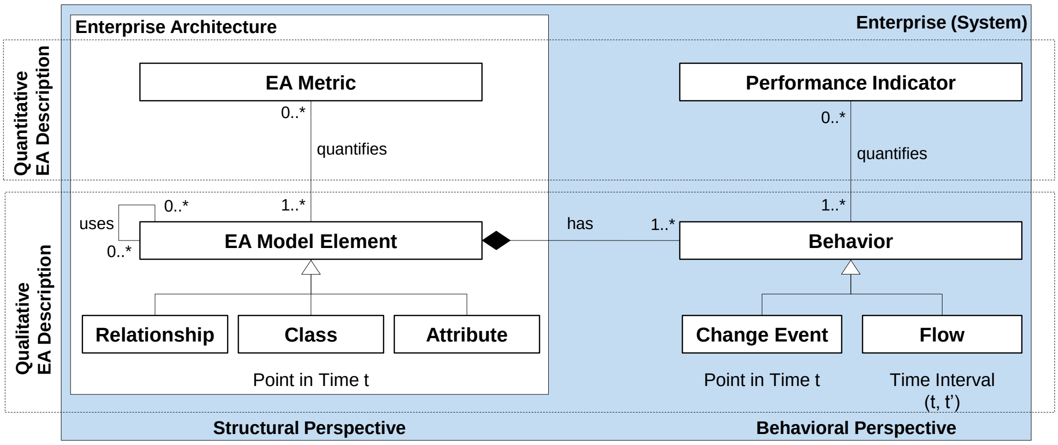


Figure 3.8.: Conceptualization of a performance indicator

Definition: Performance indicator
A performance indicator provides quantitative information regarding the (enterprise) system's behavior for a given time interval.

Additionally, a performance indicator is calculated at regular intervals. It requires the existence of a corresponding EA model, i.e., at least one EA model element. Furthermore, performance indicators focus on behavioral aspects of the underlying EA model based on change events (time-discrete) or flows for a specific period of time (time-continuous). Equipped with this definition, performance indicators support the fourth previously presented usage scenario for quantitative models in the domain of EA management, as example 3.2 shows:

Example 3.2: Example for the usage of performance indicators. To ensure the goal achievement of the EA metric from example 3.1, four performance indicators based on the given EA model are defined. Firstly, a performance indicator is required to quantify the progress of the business process criticality classification. Business processes are classified by their owners and the classification result is entered into the EA repository (at instance level of the EA model). A classification of a business process (by setting its state from unknown to critical or not critical) is to be understood as a time-discrete change event. In analogy, the performance of the business application criticality classification (performed by their owners) is quantified. These two classifications can be done in parallel, if the topology of business processes supported by business applications is known (cf. the *relies on* relationship in the model). As soon as a business application is classified as critical, a risks manager must ensure the existence of an associated IT continuity plan. The performance of this process is quantified in analogy to the previous two cases. Finally, all three performance indicators are aggregated to one overall performance indicator targeting the performance of the entire compliance requirement implementation process. Additionally, concrete plan values for the implementation process, as well as the three sub processes are defined as a plan towards the goal achievement. The continuous monitoring of these four performance indicators supports the monitoring of the progress and the early identification of problems (e.g. missing classification of a certain business process after a predefined deadline has passed) and can help to define countermeasures in a timely manner.

⌟

According to our observations in the application of quantitative models in German industry [Ma12a], we detect that business and IT stakeholders are usually interested in defining and monitoring EA metrics, whereas enterprise architects, as well as the responsible persons for the achievement of predefined the EA metric values are usually interested in defining and monitoring related performance indicators in terms of timely and reliable management decision support.

Figure 3.9 illustrates the entire terminology of our thesis. Next to the already presented terms

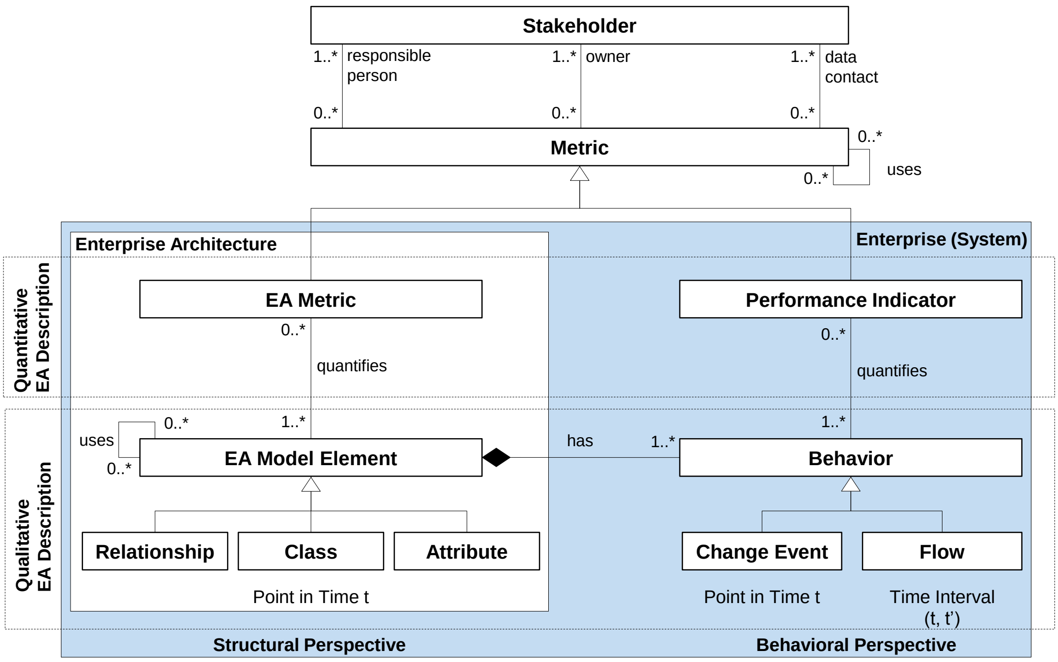


Figure 3.9.: Overview of the thesis's terminology

EA metric and performance indicator, we define the term **metric** as a generalization of these two concepts. In line with literature (e.g. [NAK02, Ad10, Pa10, St12, Kü10, Kü13, SRS10]), metrics are usually aggregated (as indicated by the *uses* reflexive relationship) to achieve a specific level of abstraction tailored to the needs of specific stakeholders. In the domain of EA management, we distinguish the following three distinct stakeholder roles in our terminology model. Thereby, every metric requires at least one from each of the following roles:

1. A **responsible person** is the person in charge for the achievement of the predefined

metric values. This stakeholder has to ensure the goal achievement by managing the underlying EA (instance) model.

2. An **owner** is interested in defining and monitoring a metric. The metric owner has to ensure that a metric (or metric system) can be implemented by providing budget for the involved stakeholder (according to literature [Kü10, Kü13] usually efforts are generated in data collection, monitoring, and reporting activities) and providing adequate organizational empowerment to the responsible person.

3. A **data contact** is a stakeholder who provides access to specific enterprise data (e.g. project portfolio management repository) or provides other enterprise data required for the computation of a given metric or metric system. As the existing literature confirms [Kü10, Kü13], enterprises should focus on an automated data collection to reduce high costs which are typical for manually collected data.

In addition to these three roles, we recommend to document an enterprise architect for each metric. Thereby, an enterprise architect can provide guidance for the metric management process and can consult the involved stakeholders. For instance, enterprise architects can identify data sources or data contacts for a specific metric and can support the definition process of metrics by providing relevant information from literature or their experience.

3.3. Risks and Countermeasures

The foundation for the identification of risks and suggested countermeasures related to the usage of metrics in the domain of EA management, was provided by Subashi as described in her Master's thesis [Su13]. Thereby, by an extensive review of the identified related work (cf. Chapter 2.3) we focus on the identification and documentation of concrete risks associated with the general usage of metrics in the investigated management disciplines as well as recommended countermeasures. As we experienced, during the first reading of the related sources, we observed that many authors describe similar risks, however, there are slightly differences in the formulations. To ensure a scientifically appropriate approach to compare and to consolidate the findings, we decide to apply a *hermeneutic* method [Bu98] for the interpretation of the studied literature. Thereby, the term *hermeneutics* originates from the ancient Greek word *hermeneuien* and means *to interpret*. Hermeneutic text interpretations are typically applied in social and behavior science and support the comparison of different information sources and the establishment of a common understanding based on text interpretations. Hermeneutics provide thereby the following two advantages according to [WRD03]:

1. text analysis in depth (beyond the obvious surface features) and

2. verification of the analysis results.

As remarked by Wallace [WRD03]) and with respect to our research interest, the texts of the related literature sources can be considered:

1. either *more* or *less meaningful*, as well as

2. either *more* or *less useful*.

To guarantee the *relevance* of our findings, we followed the recommendation of Wallace [WRD03] and apply a hermeneutic text interpretation in our research. According to Heidegger [He96], the *circular structure* of the text understanding is one of the fundamental concepts in hermeneutics. Heidegger describes the term *hermeneutic circle of understanding* as depicted in Figure 3.10. Thereby, the understanding of a given text as a whole is established by reference to its individual parts and the understanding of each individual part by a reference to the whole. Figure 3.10 illustrates that starting reading part A of a given text a certain understanding is developed. By continuing reading part B, the understanding (from the reader's perspective) expands. Consequently, a reader gains deeper knowledge about a given text by investigating the text in an iterative manner. For further details about hermeneutics, we refer to Heidegger [He96] and Wallace [WRD03].

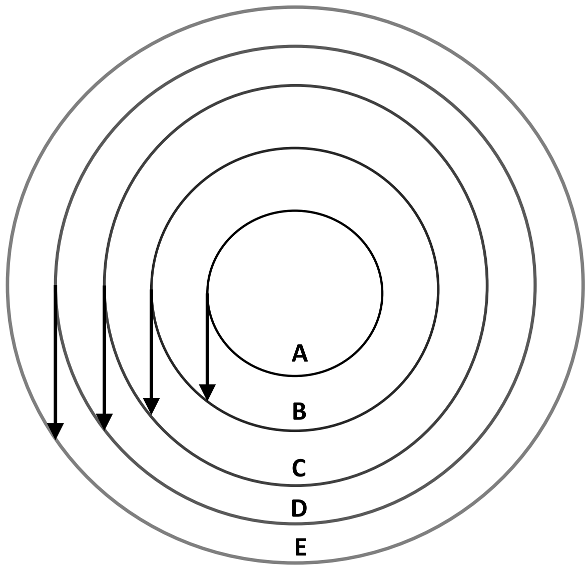


Figure 3.10.: The hermeneutic cycle according to Heidegger [He96]

As described by Subashi [Su13], the studied literature sources describe several risks related to the application of metrics in management disciplines. Additionally, the sources show that these risks depend on the organizational context and culture. Nevertheless, not all of the investigated sources (cf. Table 3.1) provided specific information about these risks and the respective countermeasures. The outcome of our research was the identification of 26 risks and 39 recommended countermeasures defined in literature as documented in [Su13]. To provide a more convenient structure regarding our findings, we defined the following nine risk categories:

1. General risks (cf. Table 3.2)
2. Data related risks (cf. Table 3.3)
3. Risks related to organizational goals (cf. Table 3.4)
4. Risks related to target values of goals (cf. Table 3.5)
5. Risks related to the number of used metrics (cf. Table 3.6)

6. Risks related to ethical aspects (cf. Table 3.7)
7. Risks related to rewards associated with metrics (cf. Table 3.8)
8. Risks related to the presentation of metrics (cf. Table 3.9)

Please note, that we slightly changed the names of the categories as well as the titles of the risks and countermeasures in this thesis, since we think the new category names are more precise than the originally published names by Subashi [Su13]. In addition, for the remainder of this section, we use the term metric according to our terminology (cf. Section 3.2) instead of the term *KPI* which was used by Subashi.

Risk	**Countermeasure**	**Sources**
Use of irrelevant metrics	Measure what is truly important, not just what is easy to measure	[HK98], [LDH08], [Pe98]
Metrics are not properly defined	Provide clear, unambiguous, and understandable definitions	[Bi05], [Bo03], [Ec10], [Ne99]
	Use a template for metrics design and provide all the necessary data	[NAK02]
Metrics are too abstract	Measure well-defined and well-designed metrics	[Ec10]
	Test metrics in advance	[Pe98]
Resistance to change	Inform the team and keep them involved	[LDH08], [Pe98], [Tu05]
Missing automation of the measuring process	Use related tools	[LDH08], [Ne02]
Standard terms	Provide clear, unambiguous and understandable definitions	[Bi05], [Bo03], [Ec10], [Ne99], [Pa10]
	Test metrics in advance	[Pe98]
	Train the involved stakeholder	[Pe98]
The metric team is not constantly informed	Inform the team about changes	[LDH08], [Pe98]
	Inform the team about the responsible person for the metric	[Pe98]
	Delegate the authority to the team	[Pe98], [LDH08]
Customer satisfaction is not measured	Define metrics that measure customer satisfaction	[Ec90], [KN91], [Ne02]
Short term focus only	Target long-term goals	[Bi05], [FGM07]

Table 3.2.: Overview of general risks and related countermeasures

Risk	**Countermeasure**	**Sources**
Considering too much (or too less) data	Appoint an analyst to scout data sources for metrics	[Ec10], [FGM07]
Defective data	Ensure data quality	[Bi05], [Ec10]

Table 3.3.: Overview of data related risks and suggested countermeasures

Risk	Countermeasure	Sources
Goal displacement	Choose metrics in alignment with organizational goals	[Bi05], [HK98], [KN91], [Pe98]
	Measure different strategic perspectives, not only financial outcomes	[KN91], [Pe98]
	Review, revise, and update metrics frequently	[LDH08], [NAK02]
Vague organizational goals	Align metrics description structure with the goal description structure	[PS10]

Table 3.4.: Overview of risks and countermeasures related to organizational goals

Risk	Countermeasure	Sources
Setting extreme target values	Interview executives and managers	[Bi05], [Ec10]
	Consider last year's targets	[Ec10]
Unchanged targets	Revise targets continuously	[LDH08]

Table 3.5.: Overview of risks and countermeasures related to target values of goals

Risk	Countermeasure	Sources
Using too many metrics	Select a minimal set of metrics	[LDH08], [St06]
	Review, revise, and update metrics frequently	[LDH08], [NAK02]

Table 3.6.: Overview of risks and countermeasures related to the number of used metrics

Risk	Countermeasure	Sources
Access to confidential data	Inform people and ask for their consent when using confidential data	[Bi05]
	Sign an agreement when the data is to be published	[Bi05]
Manipulated outcomes	Ensure data quality	[Bi05], [Ec10]
	Test metrics in advance	[Pe98]
Metric values used to punish	Use metric values to empower	[LDH08]

Table 3.7.: Overview of risks and countermeasures related to ethical aspects

Risk	Countermeasure	Sources
Delaying rewards	Reward staff as soon as appropriate	[HK98]
	Look for metrics measuring the achievement of a future state	[HK98], [LDH08]
Attach rewards to metrics too soon	Do not promise rewards too soon in the process	[LDH08]

Table 3.8.: Overview of risks and countermeasures related to rewards associated with metrics

Risk	Countermeasure	Sources
Ambiguous relationships between metrics	Use standard and consistent terminology for each relationship between metrics	[St12]
The metrics presentation causes divergent interpretations	Use standard terms	[Bi05]
	Recruit design experts and train staff	[Ec10]
	Test the effect of presentation by using prototypes	[Bi05]
Static metric structure	Design dynamic metric structure	[Ec10]

Table 3.9.: Overview of risks and countermeasures related to the presentation of metrics

To validate our findings for the domain of EA management, we designed and performed an expert survey. The survey contained two essential parts - a *background information* part concerned with the collection of data regarding the occupation and relevant working experience of the participating expert and a *risks and countermeasures evaluation and prioritization* part, concerned with the validation of our findings and with the identification of the top five risks towards the successes of EA management metrics initiatives by the experts.

Firstly, a printed version of our questions was distributed to the participants of the 8^{th} congress of the working group *Architektur Management* (in English – Architecture Management) hosted by the *Softwareforen Leipzig* [1]. The conference took place between March 4^{th} and March 5^{th} 2013. In the first conference day, we presented the identified risks and countermeasures in detail to the participants during a 45 minutes presentation slot. Thereafter, we distributed the printed exemplars of the survey (cf. appendix section of Subashi's thesis [Su13]) to every participant and asked them to fill out the sheets and to submit them anonymously to us until the end of the congress (March 5^{th}, 2013). After the given deadline was passed, we successfully collected ten full responses. In addition, we collected three partially filled responses, where the first part of the survey was completed, but the given prioritization by the experts contained only three risks (in two cases) or four risks (in one case) instead of five risks as requested by us. Lastly, we collected additional three submissions, where the first part of the survey was filled out, but no prioritization was provided by the experts. Hence, we consider only the full ten submissions and the three partially filled out submissions containing at least three or respectively four risks. Secondly, we prepared and launched an online version of our survey and invited additional enterprise architects known to our chair. The online survey took between April 3^{rd}, 2013 and April 21^{st}, 2013. After the given deadline was passed, we collected only one full response and one partially completed response, which unfortunately does not contain a risk prioritization. Therefore, in total, from both surveys we successfully collected and

[1] `http://goo.gl/q1YMLpl`

consolidated fourteen responses which we use to validate our findings as presented in the next paragraphs.

The motivation behind the questions of the *background information* part of the survey was to obtain an detailed understanding of the occupation, experience, and organizational context of the participants. In addition, we aimed to collect reliable feedback regarding the typical drivers for the usage of EA management metrics, typical stakeholder interested in the usage of EA management metrics as well as to collect concrete examples used or desired EA management metrics. This part of the survey contained following nine questions:

- *What industry branch are you working in?*
- *What is your current professional occupation?*
- *For how long have you been working in the area of EA management?*
- *For how long have you already been working with metrics in general?*
- *For how long does your company already employ EA management?*
- *Why does your organization need EA management metrics?*
- *Which stakeholders in your organization are interested in EA management metrics?*
- *What are the (expected) benefits of using EA management metrics?*
- *If your company already uses or plans to use EA management metrics, please briefly describe the three most important ones.*

The participants reported to employed in following industry branches - *finance* (8), *IT services* (2), *energy* (1), *government* (1), *consultancy* (1), and *healthcare* (1). With respect to their professional occupation, the participants reported the following results - *enterprise architects* , *IT architects* , *consultant* , *business architect*. With respect to their professional occupation, following feedback was given - *enterprise architects* (6), *IT architects* (6), *consultant* (1), and *business architect* (1). Consequently, the majority of the participants in our survey can be considered as architects, mostly employed in the financial sector.

Further, following feedback was given by the participants regarding the question of their relevant working experience in the field of EA management - *one to five years* (8), *six to ten years* (2) and *ten or more years* (1). Surprisingly, none of the participants reported *less than a year*. Additionally, the participants reported the following experience with EA management in their organizations - *less than one year* (1), *one to five years* (12), and *six to ten years* (1). Consequently, none confirmed *ten or more years* of EA management experience in their organizations. This information was complemented by the following feedback regarding the participants working experience with metrics for management purposes - *less than one year* (9), *one to five years* (4), and *six to ten years* (1). None reported *ten or more years* of relevant working experience. These findings support our understanding that EA management initiatives focus firstly on qualitative descriptions of the EA as a decision support base, and only after a certain maturity level of the EA management initiative is reached, a demand for quantitative descriptions arises.

Further, we asked the participants to indicate concrete drivers for the usage of EA management metrics in their enterprises. This question allowed multiple answers where we provided five prominent drivers according to the related work. An additional *other* option to the question was implemented to empower the participants to report drivers which were not contained in our list. According to the responses, *to support EA management decision making* and *to measure the achievement of predefined EA management goals* were reported the most prominent drivers by 6 responses. Then, *to measure the benefits (added value) of the EA management initiative* was reported by 5 responses, followed by *to justify the need for architectural projects* as reported by 4 of the responses. The list is concluded by *to improve EA management processes* as named by 3 and *risk management* as stated by 1 of the participants. Thereby, the option *other* was used three times for this question. Each of the following drivers - *risk management*, *to support projects and management decisions* and *identification of improvement potentials* was reported by 1 of the participants. According to our understanding, the driver *to support projects and management decisions* can be assigned to our predefined driver *to support EA management decision making* and consequently, the score is to be changed to 7. In analogy, the driver *identification of improvement potentials* can be assigned to our predefined driver *to improve EA management processes* and its score has to be changed to 4. The results for this question are summarized as depicted in Figure 3.11.

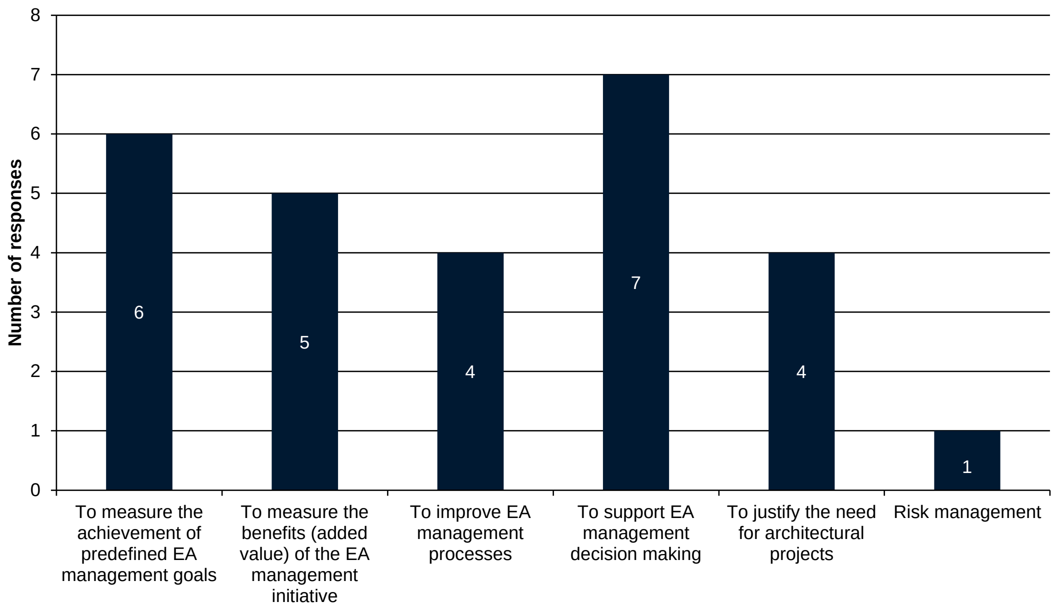


Figure 3.11.: Overview of typical drivers for the usage of EA management metrics

The next (open) question regarding the identification of stakeholders within the given organization interested in the usage of EA management metrics provided 20 heterogeneous answers. The answers are illustrated in Table 3.10. Thereby, the column *reported stakeholder* represents the exact stakeholder position as described by the participants. The column *responses* indicates the number of participants, who reported the corresponding stakeholder. To provide a more general representation of the findings, in column *consolidated term* we created a mapping to more general stakeholder terms according to our understanding, cf. Figure 3.12.

Reported stakeholder	Responses	Consolidated term
EA management leader	3	Head of EA management
Business developers	1	Business manager
Controlling	1	IT controller
Business unit managers	1	Business manager
Management	1	IT manager
IT architects	2	IT manager
IT management	4	IT manager
CFO	1	CFO
CIO	6	CIO
COO	1	COO
CxO	1	CxO
IT controlling	1	IT controller
Business management	1	Business manager
Business partners	1	Business manager
Risk management	1	Risk manager
IT security	1	IT manager
IT strategy	1	IT manager
Global Enterprise Architecture IT Management	1	Head of EA management
Business account managers	1	Business manager

Table 3.10.: Overview of typical stakeholders and a mapping to more general stakeholder terms

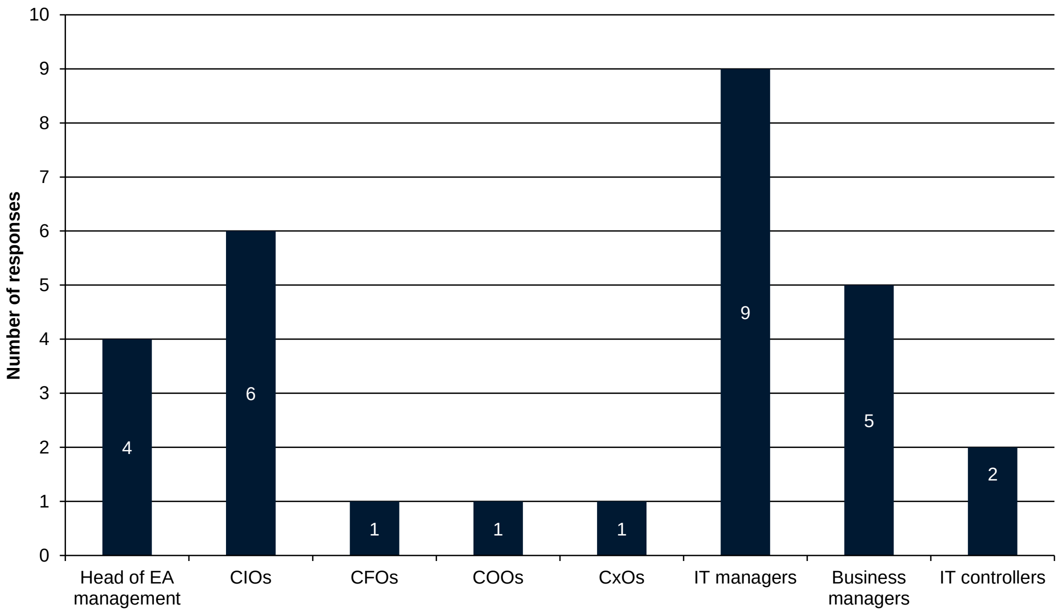


Figure 3.12.: Stakeholders interested in the usage of EA management metrics

Thereby, *IT manager* was reported by 9, *CIOs* by 6, *business manager* by 5, *head of EA management* by 4 and *IT controller* by 2 of the participants. The list is concluded by *COO*, *CFO*, *CxO*, and *risk manager*, each reported by 1 of the participants.

Asked by another open question regarding the *associated (expected) benefits of the usage of EA management metrics*, the participants provided even more heterogeneous answers as illustrated in Table 3.11.

Reported benefit	**Responses**	**Consolidated term**
Justification of the benefits provided by EAM	1	Justification of the benefits provided by EA management
Costs transparency/justification	1	Increased transparency
Identification of improvement potentials (analysis of roots for long project duration)	1	Enable identification of EA improvement potentials
Better control	1	Improved EA management controlling
EAM process improvements	1	Enable identification of EA management process improvements
Improved risk management	1	Improved risk management
Improve EAM in general	1	Enable identification of EA management process improvements
EAM Controlling	1	Improved EA management controlling
Improved alignment of IT strategy to the business strategy	1	Enable identification of EA improvement potentials
Timely identification of problems	1	Enable identification of EA management process improvements
Improved architecture controlling	2	Improved EA management controlling
To prove the benefits of EAM	1	Justification of the benefits provided by EA management
Improved decision making	2	Improved EA management decision making
Transparency	2	Increased transparency
Efficiency of EAM	1	Enable identification of EA management process improvements
Objective measurements	1	Improved EA management controlling
Comparable measurements	1	Improved EA management controlling
Landscaping	1	Improved EA management decision making
Prioritization	1	Improved EA management decision making
Strategy definition	1	Improved EA management decision making

Table 3.11.: Typical benefits associated with the usage of EA management metrics as mentioned by the participants and a mapping to more general benefits

In analogy to the question of typical stakeholders interested in the usage of EA management metrics, Figure 3.13 summarizes the consolidated responses according to our understanding.

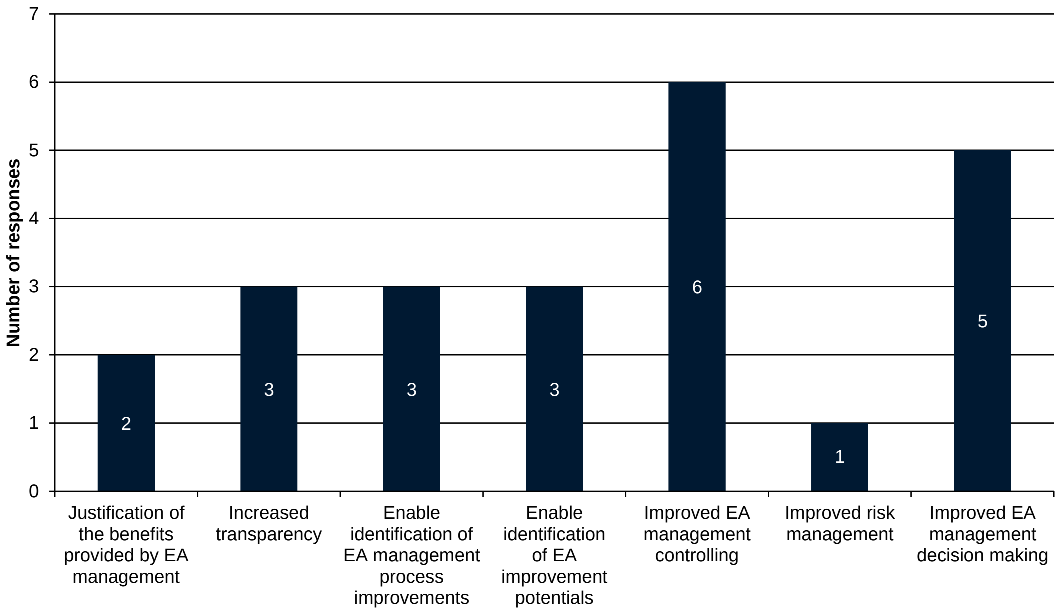


Figure 3.13.: Overview of typical benefits associated with the usage of EA management metrics

Based on the consolidation of the responses, 6 reported *improved EA management controlling* as the most relevant benefit associated with the usage of EA management metrics. Further, *improved EA management decision making* was reported by 5 and *enable identification of EA management process improvements* was reported by 3 of the participants. Each of the two benefits *enable identification of EA improvement potentials* and *increased transparency* was reported by 3 of the participants. Further, 2 confirmed *justification of the benefits provided by EA management* and 1 reported *improved risk management* as additional benefits associated with the usage of EA management metrics in their organizations.

The last (open) question of this first part of the survey was concerned with the identification of concrete metrics used or desired to be used for the EA management initiatives of the participants' organizations. As expected, the participants provided very heterogeneous metric examples as depicted in Table 3.12. As the collected data confirms, different personal understandings regarding the term metric exist. In our understanding, none of the example precisely describes a metric. Moreover, the majority of the examples, e.g. "Data usage monitoring", "Costs and EA life cycle" do not provide any information regarding a concrete computation rule, required data and a description of an associated model and organizational goals. This insight is in line with the results of the previous question targeting the working experience with metrics in management disciplines, in which the majority of the participants (9/14) reported less than one year. Nevertheless, as the collected responses confirm, the typical EA elements - business applications and costs, as well as the typical EA cross-cutting concerns standards and projects seem to be objects of broader interest for the usage of quantitative models.

Reported metric	Number of responses
Data quality assessment	1
Data usage monitoring	1
Data users identification	1
Number of applications in the latest EA lifecycle phase	1
Operations costs of business applications	1
Number of standardized software products vs. number of total software products	1
Complexity	1
Conformity to standards	1
Costs	1
EA lifecycle	1
Complexity vs personnel skills	1
Criticality	1
Managed evolution	1
Business to IT alignment	1
Landscape complexity reduction	1
Cost reduction	1
Time to market of IT projects	1

Table 3.12.: EA management metric examples provided by the participants and a corresponding mapping to typical EA elements

In the second part of the survey, we focused on the evaluation of the correctness and completeness of the identified risks and countermeasures as well as the prioritization of the five most critical risks towards a successful EA management metric initiative. Therefore, for each risk and each associated countermeasure, we provided one prioritization question (a closed question) and one comment field (an open question) in the survey. In the prioritization field, a participant could indicate by placing an 'X' character, that the corresponding risk is critical towards the success of an EA management metrics initiative based on the participants working experience. Thereby, every participant was asked to mark accordingly five of the presented risks. Figure 3.14 summarizes the prioritization results based on the 14 responses.

Please note, that six risks and their corresponding countermeasures which have been selected by none of the participants are not included in the listing - *negotiated goals*, *access to confidential data*, *legal limits*, *delaying rewards*, *rewards attached to metrics too soon*, and *static metric structure*.

In the comment field to each risk and countermeasure, the participants could provide feedback whenever the associated risk/countermeasure is described well, is not relevant, or is wrong. To our surprise, all participants agreed with the correctness and completeness of the provided list, i.e., no risks/countermeasures have been reported as wrong, irrelevant or not suitable for the for the domain of EA management. We also received two comments regarding two specific risks. In the first one, a participant highlighted the importance of the understanding and documenting relationships between metrics as suggested by our corresponding countermeasures and fully agreed with the validity of the related risk regarding the criticality of making metrics

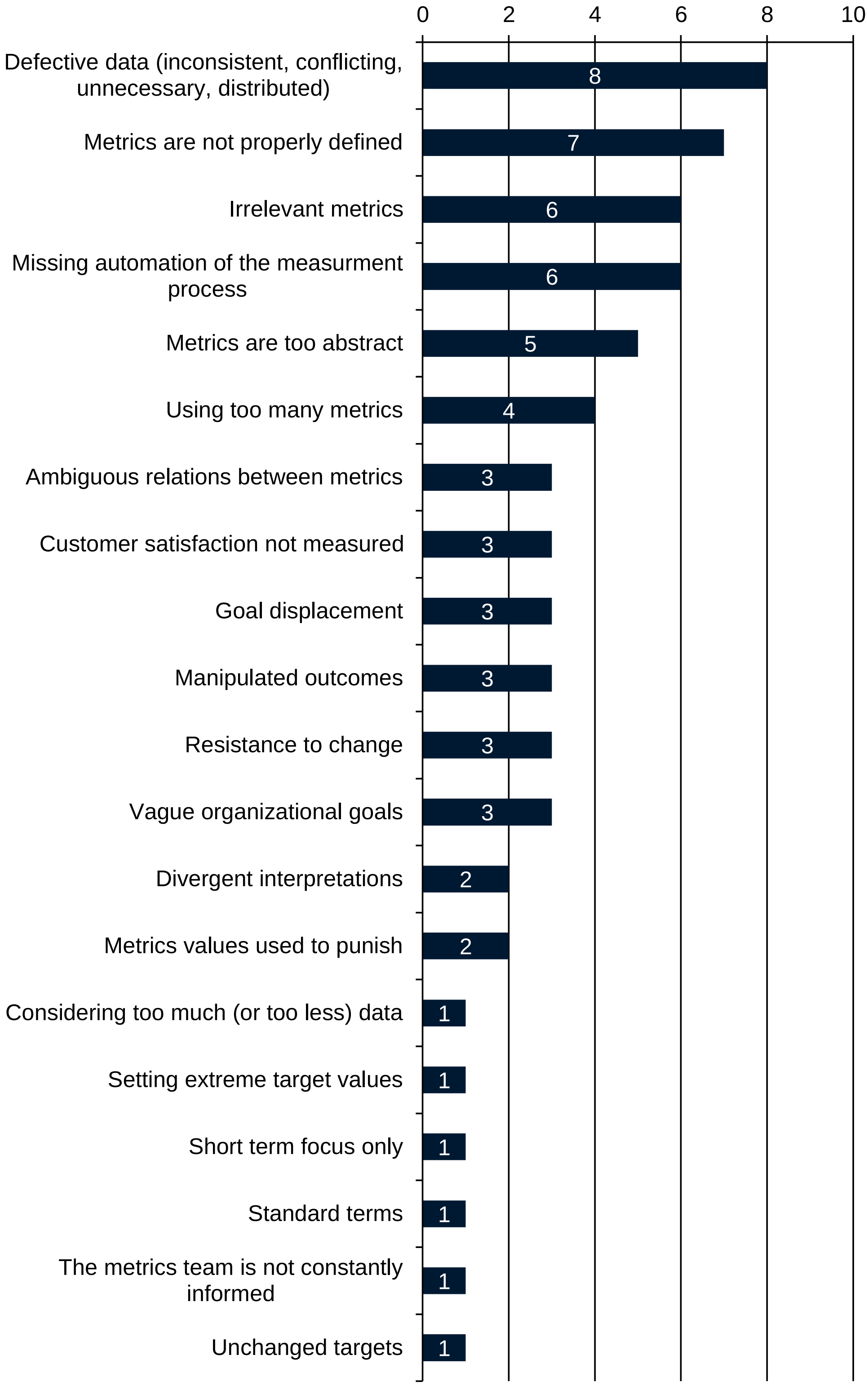


Figure 3.14.: Results of the risks prioritization

aggregations transparent. In the second one, another participant highlighted the relevance of the risk of missing alignment between EA management metrics to the organizations strategy and emphasized the importance of the suggested countermeasure - revising and updating all metrics whenever the strategy and the corresponding goals are changed.

In the very last question of the second part of the survey the participants could provide risks and countermeasures according to their experience, which are not contained in our lists. However, we did not receive any additional input, thus, we consider that our findings are complete with respect to the working experience of the participants. According to our results as depicted in Figure 3.14, the five most highly prioritized risks are:

1. Defective data
2. Metrics are not properly defined
3. Irrelevant metrics
4. Missing automation of the measurement process
5. Metrics are too abstract

Based on Subashi [Su13], these risks and the suggested countermeasures can be summarized as follows. According to Eckerson [Ec10], *inaccurate and untrustworthy data* is the main cause of damaging the credibility of a measurement initiatives, since such data can be *extremely misleading.* To address this issue, Eckerson [Ec10] and Bird et al. [Bi05] suggest that measurement initiatives must consider more attention on *data quality.* According to Bird et al., firstly the data quality for the computation of a metric must be determined and secondly it must be ensured, that the collected data is not been corrupted (e.g. intentionally manipulated by the responsible person) and to examine and isolate suspect data, which is collected manually.

Secondly, as confirmed by [Bo03, Ne99, Pa10], metrics are usually not properly defined. On the one hand, a missing terminology base for the involved stakeholders in the metrics measurement process can lead to misunderstandings and to different interpretations. On the other hand, the quality of terminology documentations is usually not good e.g. it contains inconsistencies. To address these issues, [Bi05, Ec10] suggest that the involved stakeholders must firstly ensure and establish a common understanding of the related concepts and models before starting with the definition and measurement of concrete metrics. In addition, Neely et al. [NAK02] suggest, that a generic and accepted metric description structure should be applied to document metrics to support the comprehensive understanding of all involved stakeholders.

Thirdly, metrics are used to measure the fulfillment and progress towards the achievement of specific organizational goals. However, sometimes, metrics used by organizations do not properly reflect the reality in these organizations [Pe98]. According to [Pe98, HK98], organizations often are interested to have "good statistics" and the relevance of the employed metrics as well as correctness of the underlying model is not systematically evaluated. According to Hauser [HK98], managers are prone to seek for metrics that can be measured precisely, nevertheless this alone does not guarantee the relevance and validity of metrics. To account for this issue, Hauser [HK98] suggests to evaluate and ensure the validity of metrics for their purpose, i.e., to evaluate and prove that the metrics are based on correct models of the reality even if the definition and measurement of these metrics is considered as difficult.

Fourthly, according to Lawson [LDH08] and Neely et al. [Ne02], the metrics measurement process includes several activities from data collection to results computation and their presentation to the corresponding stakeholders. However, in many organizations the process of data collection is usually done manually (e.g. by filling out Excel spreadsheets). Thereby, several problems can occur, e.g. input of (intentionally or non-intentionally) wrong data, usage of wrong computation rule of a given metric, or wrong reporting of the results. To mitigate these risks, Lawson [LDH08] suggests to automate the process of data collection, metrics calculation, and results reporting by using specialized tools or tool chains for this purpose.

Fifthly, according to Eccles [Ec90], metrics are usually described too abstract. Hence, the understanding and implementation of these metrics is ambiguous. More precisely, the description of metrics often does not clearly describes the data required for the computation of the metrics, does not provide an unambiguous description of the underlying calculation rule and model, or does not specify certain metric properties, e.g. target values or measurement frequency. To address this issue, Eccles [Ec10] suggests to implement and measure only "well-defined" and "well-designed" metrics. In addition, the author suggests to test metrics in advance (right after their definition and before their implementation) to validate required data sources and data quality, as well as to get feedback from the involved stakeholders as early as possible. Neely et al. [NAK02] further suggest to define and to use a uniform metric description structure to support the common understanding of different stakeholders.

Additionally, in several discussions with the experts during conference breaks, the following risk was discussed quite frequently. According to Perrin [Pe98] and Lawson [LDH08], organizations can face the occurrence of the so-called *meeting the numbers* phenomenon (according to [LDH08], this phenomenon is associated with our risk category *goal displacement*). Thereby, managers responsible for the achievement of specific target values for metrics are aware, that a certain metric is irrelevant for its purpose, however achieving the target values provides financial benefits or the not achieving the target values leads to financial punishment. Thus, managers are prone to perform economically not reasonable actions just to ensure the achievement of predefined target values. To mitigate this risk, Lawson [LDH08] suggests next to the usage of well-defined metrics and correct underlying models, a clear alignment of metrics to concrete strategic goals of the organization. As Lawson further states, strategies needs to be updated over the time. Therefore, related metrics must be revised whenever an update of the organizations strategy is performed to ensure that the metrics are still aligned with the organizational goals, and if required, to delete not anymore needed or to define new metrics.

3.4. Metric Management Fact Sheet

According to the related literature, the usage of a well-defined metric description structure is an important part of the metrics definition process. According to Neely et al. [NAK02], in practice, people usually forget to think about or to document several important metric aspects. Further, many of the risks presented in Section 3.3 can be mitigated, if a comprehensive and well-defined metric description structure is used. Hence, in this section, we present a generic and uniform metric fact sheet for the domain of EA management. Therefore, we firstly study the identified related literature (as described in Chapter 2) with the research interest for the identification of proposed metric description structures. In our understanding, the identified description structures, and their suggested elements, serve as an starting point towards the definition of the targeted metric management fact sheet. Hence, the next three subsections provide an overview of our findings in the related sources from the investigated three management fields.

3.4.1. Metric Fact Sheets in IT Management Literature

To identify concrete description elements for EA management metrics, we firstly examined the literature sources from the field of IT management (cf. Table 2.3). As we observed, only Kütz describes a structure relevant for our research interest in his book [Kü10, p. 47] as depicted in Figure 3.15.

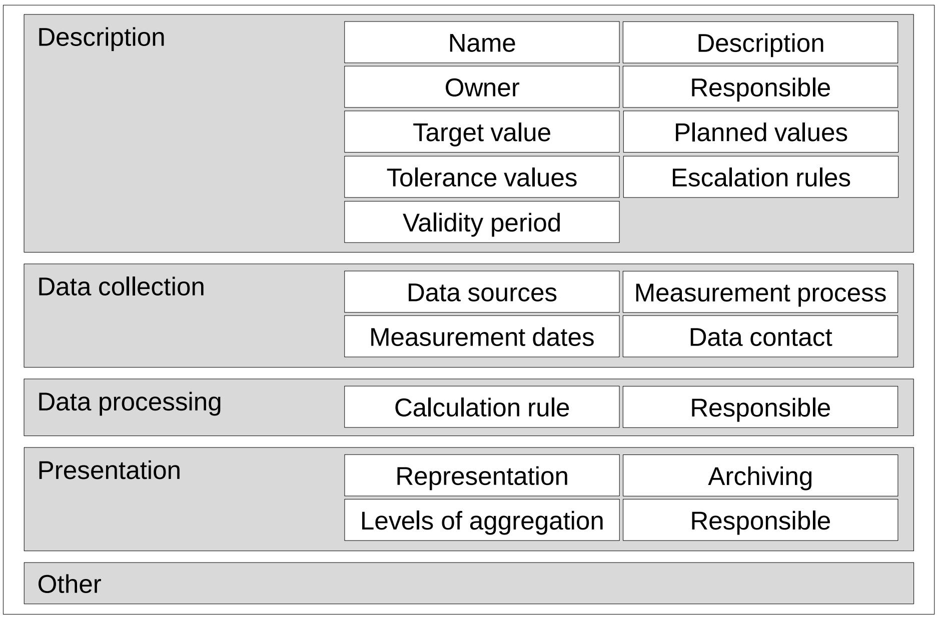


Figure 3.15.: IT controlling metric description structure according to Kütz [Kü10]

The proposed structure consist of 19 elements organized in five categories as follows (please note, that we translated all of the elements and categories from German and aligned the translation with our terminology, cf. Section 3.2):

- The first category *description* is concerned with the documentation of nine metric description elements to ensure comprehensive understanding of the intended metric usage for all involved stakeholders. Firstly, a *name* element is proposed by the author with the purpose to ensure the timely recognition of metrics and to foster intuitive understanding of the measurement purpose. Secondly, a *description* element concerned with the documentation of the motivation and the intension regarding the usage of a metric (in natural language) is proposed. Thirdly, an *owner* and a *responsible* person for the achievement of specific metric values are proposed. Fourthly, one *target* value, (optionally) several *planned*, and (optionally) several *tolerance* values for a metric are recommended for documentation. Thereby, tolerance values describe allowed deviations to predefined planned and target metric values. Fifthly, *escalation rules*, providing behavior guidance for the responsible person for the case that specific metric values are reached or specific empowerment constraints are not fulfilled, should be documented. Lastly, the *validity period* (the intended time interval for the usage of a metric) is proposed for documentation.

- The second category - *data collection* - is concerned with the identification and the access to data sources required for the computation of a metric. Hence, the element *data sources* is concerned with the documentation of concrete data sources containing specific information in a given organization, e.g. risk management repository or project management repository. The two elements *measurement process* and *measurement dates* are concerned with the documentation of the intended measurement frequency. Finally, a *data contact* is to be defined and documented for each source. This stakeholder is responsible for ensuring the availability and specific quality of the required data.
- The third category - *data processing* - is concerned with the calculation of a metric. Therefore, firstly, the element *calculation rule* is concerned with the documentation of the underlying calculation of the metric. Secondly, a stakeholder responsible for the execution of the predefined calculation rule is (optionally) documented.
- The forth category - *presentation* - is concerned with the documentation of different metric aspects related to the (graphical) presentation of the computed values. For example, the representation of a metric can contain current, planned, target, and historicization metric values. Further, the element *archiving* is proposed to document the archiving strategy of the measured metric values. The element *levels of aggregation* is proposed to document (eventually) performed aggregations applied to the metric results to support stakeholder-specific representations. Lastly, a *representation responsible* person is proposed for documentation. This stakeholder is responsible to ensure the implementation and provision of the predefined metric representations.
- The fifth category - *other* - is proposed as a generic support for the documentation of additional organization-specific metric details, which cannot be attributed to any of the previously introduced description elements.

3.4.2. Metric Fact Sheets in Enterprise Management Literature

After reviewing the related literature from IT management, we continued our research with examining the sources from our enterprise management literature pool (cf. Table 2.4). As we found out, three of the sources - (Ne02, Pa10, Po10) - propose concrete management metric description structures, which we present in the subsequent paragraphs.

Parmenter describes in his book [Pa10, p. 263-264] a metric description structure consisting of ten elements as illustrated in Figure 3.16. However, he only names these elements and does not provide any information regarding their purpose. Hence, in our understanding, several of the elements can be considered ambiguous and allow different interpretations from a reader's perspective. Thus, in the following description of Parmenter's elements we describe our understanding of the proposed elements. Please note, that we replaced the term *performance measure* as used by Parmenter by the term *metric* according to our terminology in the subsequent citations.

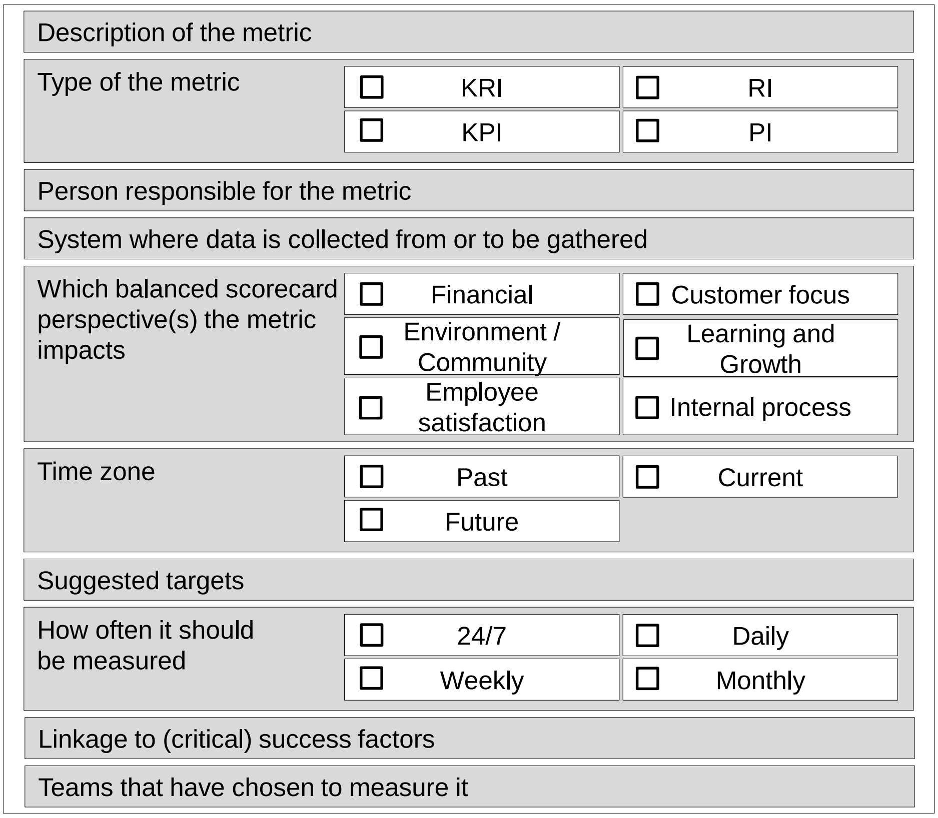


Figure 3.16.: Management metric description structure according to Parmenter [Pa10]

- *Description of the metric.* In our understanding, this element can be considered as similar to the *description* element proposed by Kütz [Kü10]
- *Type of the metric (KRI, RI, PI, KPI).* In contrast to our terminology, Parmenter distinguishes between four types of metrics - *result indicators* (RI), *key result indicators* (KRI), *performance indicators* (PI), and *key performance indicators* (KPI). According to the author, every metric must be classified as an instance of exactly one of these four types.
- *Person responsible for the metric.* In contrast to Kütz, Parmenter does not distinguish between separate responsibilities for data collection, computation, presentation, and the achievement of predefined metric values. In our understanding, this definition is ambiguous and hence we map this term to our term *responsible person* (cf. Section 3.2).
- *System where data is collected from or to be gathered.* In our understanding, this element is concerned with the documentation of the data sources required for the calculation of the metric. Surprisingly, the author speaks of a "system" instead of "systems". However, usually (cf. [Kü10, NAK02]), several different data sources are required for the computation of metrics and hence, we understand this element as a documentation of several related information sources.
- *Which balanced scorecard perspective(s) the metric impacts.* In our understanding, this

element is concerned with the documentation of balance scorecard perspectives, which are quantified by a metric. Thereby, in contrast to Kaplan and Norton [KN91], Parmenter uses a BSC consisting of the following six, instead of the four classical BSC perspectives (cf. Section 2.3.3) - *financial*, *customer focus*, *environment/community*, *internal process*, *employee satisfaction*, and *learning and growth*.

- *Time zone (past, current, future)*. We understand this element as a classification of the measured controlling object state (cf. Section 2.2). Thereby, a metric can be used to quantify either a past state, the current state, or an expected future state of a controlling object.
- *Suggested targets*. In our understanding, this element tackles the documentation of target and planned metric values.
- *How often it should be measured (24/7, daily, weekly, monthly)*. In our understanding, this element is concerned with the documentation of the intended measurement frequency.
- *Linkage to (critical) success factors*. In our understanding, this element is used to document so-called *(critical) success factors* related to a metric. According to Parmenter [Pa10, p. 199], critical success factors are defined as “a list of issues or aspects of organizational performance that determine ongoing health, vitality, and wellbeing”.
- *Teams that have chosen to measure it*. According to our understanding, different teams within a given organizations can decide if they want to use a specific metric.

Popova et al. present in [PS10, p. 11-12] a management metric description structure comprising nine distinct elements, cf. Figure 3.17. In analogy to Parmenter, two of these elements are

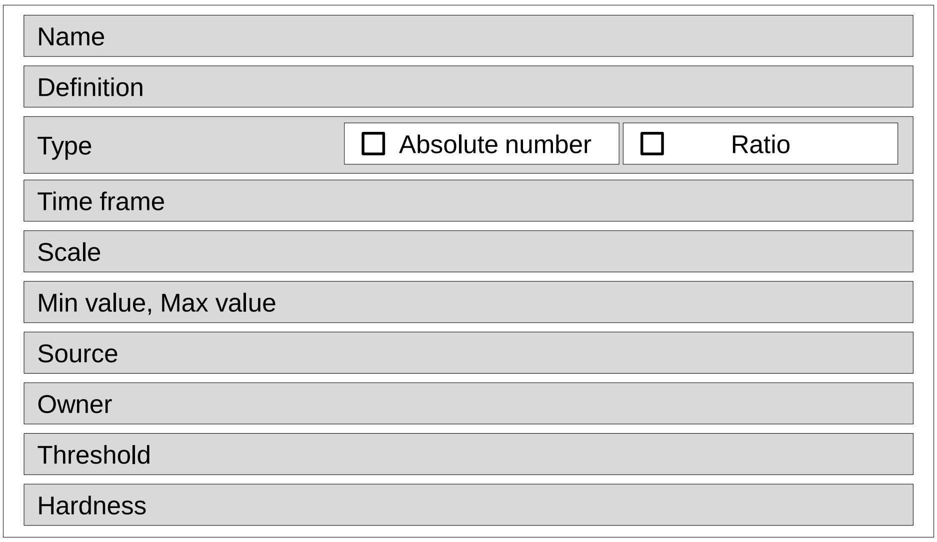


Figure 3.17.: Management metric description structure according to Popova et al. [Pa10]

not explained by the authors, thus we describe these elements based on our understanding of the metric examples provided by the authors. Please note, that we replaced the term *indicator* as used by Popova et al. by the term *metric* according to our terminology in the subsequent citations.

- The element *name* is only stated by the authors and a description for its purpose is not given. We understand this element as a title of a metric.
- The element *definition* is also only named by the authors. Based on the provided metric examples by the authors, we understand this element as a description of the calculation rule of a metric.
- A *type* of a metric is described as "continuous or discrete, for example if the metric can be measured as a continuous number then its type can be specified as continuous and if the metric is measured in indivisible units such as packets, items, pieces or using predefined concepts such as low/medium/high then it can be specified as discrete". In our understanding, this element corresponds to the classification of a metric as an *absolute number* or *ratio* (cf. Section 2).
- The element *time frame* is defined as "[..] the length of the time interval for which it (the metric) will be evaluated, e.g. the metric 'yearly profit' has time frame year, 'number of customers per day' has time frame day". In our understanding, this element is concerned with the documentation of the intended measurement frequency of a metric.
- A *scale* of a metric is defined as "[..] the measurement scale for the metric, different scales can be predefined and referred to here".
- The element *min value, max value* is defined as "[..] when a predefined scale is used and only a part of this scale is relevant for the particular metric".
- A *source* is defined as "[..] which was the internal or external source used to extract the metric: company policies, mission statements, business plan, job descriptions, laws, domain knowledge, etc. – these sources contain (informal) statements about the desired state or behavior of the company and regulations it has to obey". In analogy to our remark regarding Parmenter's source concept, we understand this element as a set of different data sources containing the required data for the computation of a metric.
- An *owner* of a metric is defined as "[..] the performance of which role or agent does it measure/describe". In our understanding, this definition is ambiguous, thus we suggest to understand this term as the responsible person according to our terminology (cf. Section 3.2).
- A *threshold* of a metric is defined as "[..] the cut-off value separating changes in the value of the metric considered small and changes considered big; used to define the degree of influence between metrics. [..] depending on the scale of measurement of the metric, the threshold can have a clearly-defined unit of measurement (e.g., measured in hours, km, number of persons or products, etc.) or it can be measured in unnamed units for qualitative scales such as low-medium-high where one unit represents the difference between two consecutive points on the scale, for example between low and medium or between medium and high (which are assumed to be equidistant)".
- Finally, the element *hardness* is described as "[..] a metric can be soft or hard where soft means not directly measurable, qualitative, e.g. customer's satisfaction, company's rep-

utation, employees' motivation, and hard means measurable, quantitative, e.g., number of customers, time to produce a plan".

Neely et al. present in their book [NAK02] a generic management metric description structure consisting of ten elements as illustrated in Figure 3.18. Please note, that we substituted the

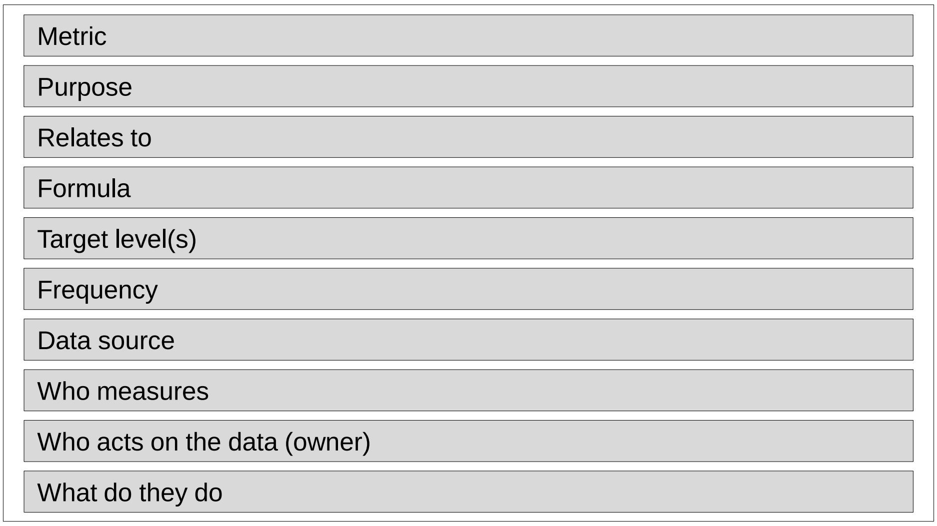


Figure 3.18.: Management metric description structure according to Neely et al. [NAK02]

term *measure* as used by the authors by the term *metric* according to our terminology in the subsequent paragraph.

- The element *metric* is concerned with the naming of the metric. Thereby, the name summarizes the purpose of a metric, and should be clearly understood by all involved stakeholders.
- The element *purpose* documents the motivation and intension for the usage of the metric and describes which type of stakeholder behavior should be encouraged.
- The element *relates to* documents other metrics, to which a given metric is interrelated and describes which initiatives or strategies are supported (quantitatively assessed) by the metric.
- The element *formula* is concerned with the documentation of the underlying computation rule of the metric. Thereby, natural and formal languages, e.g. mathematical equations, can be used. According to the authors, this element must additionally clearly describe the data sources required for the intended calculation.
- The element *target level(s)* is concerned with the definition of target and planned values.
- The element *frequency* is concerned with the documentation of the intended measurement and reporting frequencies.
- The element *data source* is concerned with the documentation of different data sources required to provide the data for the calculation of the metric.

- The *who measures* element describes in our understanding the concept of owner according to our terminology (cf. Section 3.2).
- The *who acts on the data (owner)* element described in our understanding the concept of a responsible person according to our terminology (cf. Section 3.2).
- Finally, the element *what do they do* is concerned with the documentation of possible actions, which the owner of a metric can perform to ensure the achievement of predefined target or planned metric values.

3.4.3. Metric Fact Sheets in EA Management Literature

We further studied the related sources from the domain of EA management with respect to our research interest. As we found out, the five sources (Fe09, Gr09, St09, Ad10, St12) describe relevant metric description structures as described below.

Firstly, Feldschmid describes in his Master's thesis [Fe09a] the need of a generic EA management metric description structure. As a possible solution for this problem, he suggests the usage of Kütz's [Kü10] structure. However, Feldschmid does not present any changes, adaptations or improvements of Kütz's structure, hence we consider there is no additional relevant knowledge provided by the author with respect to research question 3.

Secondly (and also inspired by Kütz), Gringel presents in his Master's thesis [Gr09] a description structure for EA metrics concerned with the assessment of business applications. As the references provided in [Ad10, p. 72] confirm, Gringel wrote his thesis under Addicks' supervision. Thus, we provide only the description of the structure described by Addicks [Ad10], which consists of the following nine elements (cf. Figure 3.19):

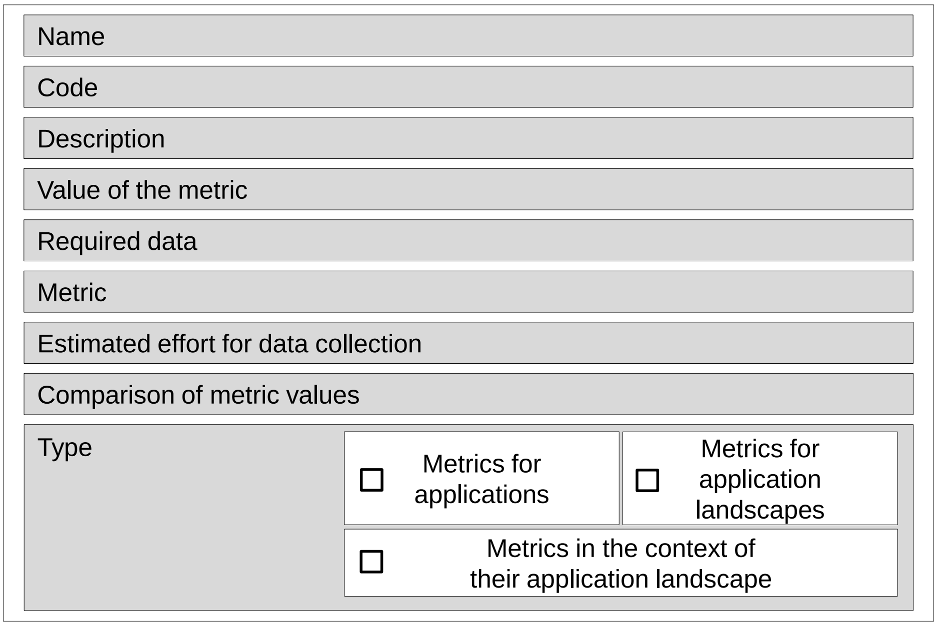


Figure 3.19.: EA metric description structure according to Addicks [Ad10]

- A *name* element is proposed to document a short and sound name of a metric.
- A *code* element is suggested to document a short unique identifier for a metric.
- A *description* element is proposed for the documentation of the intension, motivation and assumptions associated with the usage of a metric.
- A *value of the metric* element is concerned with the documentation of different assessment criteria, which can be quantified with a metric.
- A *required data* element is proposed for the documentation of the sources containing the data required for the metrics calculation.
- The element *metric* provides a semi-formal description (natural language in combination with mathematical equations) of the calculation rule of a metric.
- The element *estimated effort for data collection* is concerned with the documentation of the expected effort for data collection of the required data.
- The element *comparison of metric values* documents how different measured values of a metric can be compared.
- Finally, the element *type* is concerned with the classification of the metric regarding the following three categories - *metrics for applications*, *metrics for application landscapes*, and *metrics in the context of their application landscape*.

As previously described in Section 2.3.3, Stutz provides in his PhD thesis [St09b, p. 110] a BSC based description structure for EA management metrics as depicted in Figure 3.20. Based on the results from his extensive literature review, the author presents a meta-model for the description of EA management metrics consisting of 36 concepts and even more relationships between these concepts. Nevertheless, the author presents in the Appendix of his work 15 exemplary metrics developed during the evaluation of his method with different industry partners. Thereby, Stutz confirms, that a description structure for his metrics must not contain all of the meta-model elements and presents a simplified description structure containing only 12 description elements as illustrated in Figure 3.20 (please note, that we translated the subsequent description from German and thereby used our term *metric* for the term *Kennzahl* as used by Stutz):

- The element *title* is proposed to document a short and sound name of the metric (three to five words).
- The element *code* is proposed to assign and document a unique identifier to a metric.
- The element *description* is proposed to support a comprehensive textual documentation of the motivation and intension for the usage of the metric.
- The element *assessment perspective* is used to document BSC perspectives quantified by the metric ("assets", "services","processes", and "finances"). These assessment criteria correspond to the four BSC perspectives as proposed by the author (cf. Section 2.3.3).
- The element *scope* is used to document the EA layers affected by a metric (in the understanding of the author, the overall EA structure consists of the following five EA layers - *strategy*, *process*, *integration*, *software and data*, and *infrastructure*).

Title		
Code		
Description		
Assessment perspective	☐ Assets	☐ Processes
	☐ Services	☐ Finances
Scope (EA layers)	☐ Strategy	☐ Process
	☐ Integration	☐ Software and data
	☐ Infrastructure	
Calculation rule		
Measurement unit	☐ Absolute number	☐ Ratio
Target value		
Related success factors		
Value drivers		
Measurement frequency		
Owner		
Remark		

Figure 3.20.: BSC-based description structure for EA management metrics according to Stutz [St09b]

- The element *calculation rule* is proposed for the documentation of the calculation of a metric. Thereby, Stutz suggests the usage of formal mathematical equation as a description of calculation rules.
- The element *measurement unit* is proposed for the documentation of the type of a metric in terms of an *absolute number* or a *ratio* (cf. Section 2.2).
- The element *target value* is concerned with the documentation of organization-specific target values of a metric.
- The element *related success factors* is proposed for the documentation of so-called *success factors* towards the achievement of the predefined target values.
- The element *value drivers* is proposed to document "[..] factors, which positively affect the company value".
- The element *measurement frequency* is used for the documentation of the intended measurement frequency of the metric.
- The element *owner* is proposed for the documentation of the responsible person according to our terminology (cf. Section 3.2).
- Finally, the element *remark* is proposed to document additional organization-specific metric information if required (as textual descriptions).

Surprisingly, Stutz does not support the documentation of data sources required for the computation of his metrics. In addition, the presented structure does not reflect the need of a *data contact* as suggested by [Kü10, NAK02].

In their DSL *MetricML*, Strecker et al. [St12] present a meta-model for the description of metrics. This model consists of 17 classes (comprising 32 attributes) and 21 relationships between the classes (cf. [St12, p. 18]). The concept metric (in the terminology of Strecker et al. "indicator") comprises the following 13 class attributes (cf. Figure 3.21):

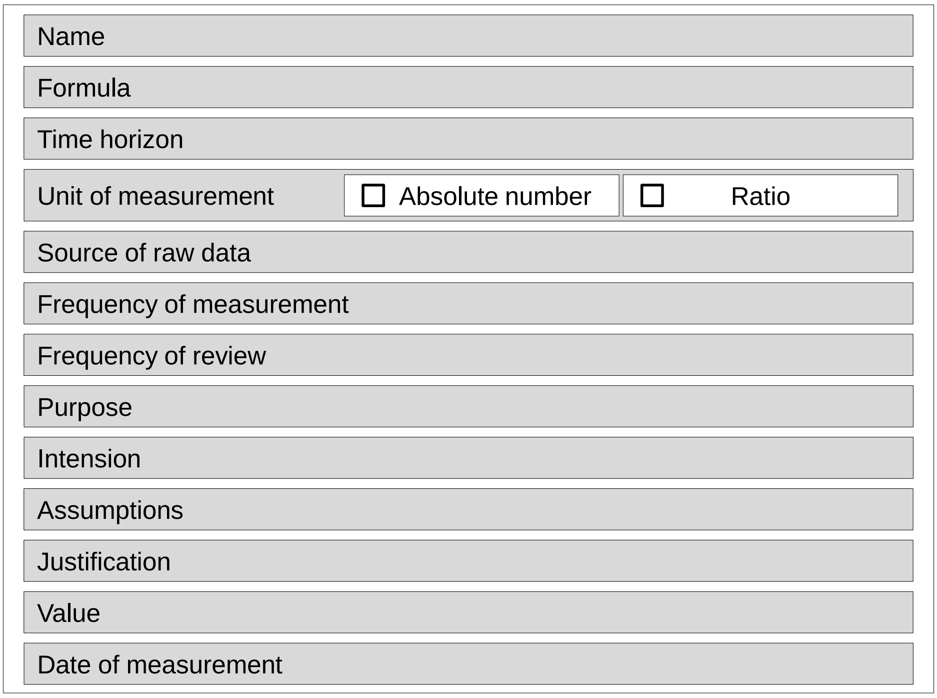


Figure 3.21.: Metric description structure according to Stecker et al. [St12]

- A *name* attribute is used to document the name of a metric.
- A *formula* attribute is used to document the computation rule of a metric.
- A *time horizon* attribute is used for the documentation planned and target values of a metric.
- The attribute *unit of measurement* is used to document the type of a metric in terms of an *absolute number* or a *ratio* (cf. Section 2.2).
- The attribute *source of raw data* is proposed for the documentation of the data sources required for the computation of a metric.
- The attributes *frequency of measurement* and *frequency of review* are used for the documentation of the intended measurement and review frequencies of a metric.
- The attribute *purpose* is used for the documentation of the purpose in respect to the usage of a metric.

- The attributes *intension*, *assumptions* and *justification* are proposed to correspondingly document the intension, justification and assumptions related to a metric.
- The “intrinsic” (cf. [St12]) attributes *value* and *date of measurement* are used for the historicization of the measured metric values. Hereby, after each measurement a pair of measurement date and measured metric value can be archived on metric instance level.

In addition, the MetricML DSL allows the documentation of related organizational goals, different stakeholders in terms of owner, responsible person and data contacts, as well as the documentation of related EA layers, e.g. business processes, organizational units or resources.

As our results show, several authors define different metric description structures in the investigated management fields. Thereby, some specific description elements, e.g. name, calculation rule, and a responsible person are documented in almost all of the investigated structures. However, some other elements, e.g. goals or a data contact are contained only in a small number of the structures. Based on these findings, in the subsequent section we present a generic EA management metric description structure which we name a ***Metric Management Fact Sheet*** **(MMFS)**.

3.4.4. A Generic Metric Management Fact Sheet (MMFS)

As an answer to our research question 3, in this section we present a generic structure designed to document and enable the organization-specific implementation of metrics in the EA management domain. For the design of our MMFS, we consider all of the proposed metric description elements by the related literature and we additionally incorporate our experience and observations during collaborations with industry partners. Further, the MMFS design is guided by the goal of proposing a minimal number of description elements, which are required to ensure a comprehensive metric documentation as starting point for an organization-specific metric implementation. The resulting MMFS structure, originally published by us in 2012 [Ma12c], is illustrated in Figure 3.22 and consists of ten description elements organized in two categories.

We want to emphasize the importance of distinguishing between two types of metric description elements - **general structure elements** and **organization-specific structure elements** (cf. Figure 3.22).

1. Several of the identified elements in literature can be classified as independent from a given organizational context, e.g. the name of a metric, or the calculation rule of a metric. As stated by [NAK02, Kü10, St09b, St12], metrics can be used for the purpose of benchmarking different organizations or organization units. Hence, a clear, unambiguous, and uniform description of these elements is required. We refer to this type of metric description elements as *general structure elements*.
2. According to the related literature, several metric description elements relate on a specific organizational context, e.g. the owner of a metric, the data contact, the predefined target and planned values. As we experienced during discussions with industry experts, such information is considered sensible and thus, should not be part of a benchmarking or subject of communication outside of a given organization (e.g. concrete names of

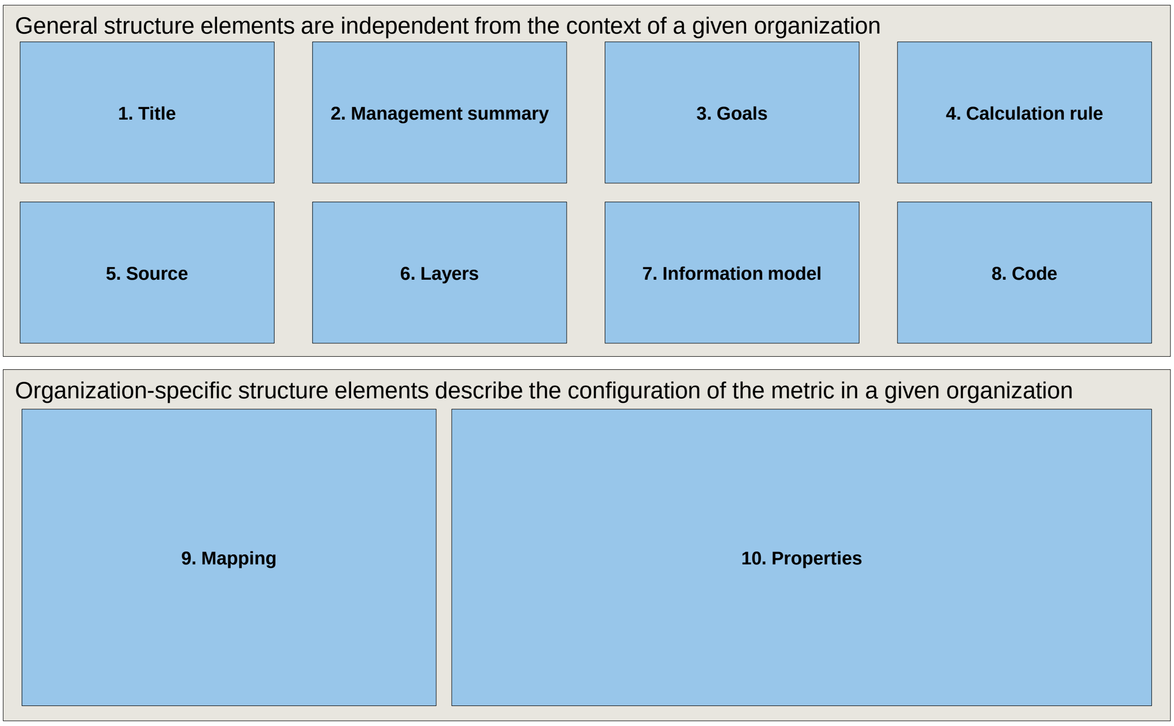


Figure 3.22.: Overview of the MMFS and its elements

employees, or concrete planned/target values of a given metric). We refer to this type of elements as *organization-specific structure elements*.

For the category *general structure elements* we propose the following eighth elements:

1. According to our understanding, every EA management metric requires a **Title** documented in a natural language. This element supports a clear, sound and as short as possible summary of the purpose of the metric. The definition of the title should be understood as a collaborative process performed by all involved stakeholders. Additionally, the documentation of a title supports the retrieval (search) of metrics.

2. Every EA management metric requires a comprehensive description of its motivation, expectation, intension, assumption, and justification for its usage. Thus, we propose the element **Management summary** as part of our MMFS. This element should be collaboratively developed and its correctness as well as completeness should be agreed by all involved stakeholders. The benefit provided by this element is the support of a common understanding of the motivation for the usage of a metric.

3. The element **Goals** - is concerned with the documentation of related EA management goals, the achievement of which is quantified by the metric. The link between a metric and goals is essential for the definition of any metric as emphasized by Basili et al. [BCR94]. Thereby, for the domain of EA management, we account for the most common EA management goals. In our work, we stick to the ten EA management goals as presented by Buckl et al. [Bu10] - *reduce operating cost*, *increase disaster tolerance*, *reduce security breaches*, *ensure compliance*, *increase homogeneity*, *improve project ex-*

ecution, *enhance strategic agility*, *improve capability provision*, *foster innovation*, and *increase management satisfaction*.

4. The element **Calculation rule** is concerned with the documentation of the exact calculation of an EA management metric. Hence, this element must contain all relevant concepts (and agreed exceptions) required for the calculation of a metric. In the literature we observed both—the usage of natural languages, as well as the usage of formal languages for the documentation of calculation rules. In our understanding, an organization interested in the usage of EA management metrics should firstly use a mixture of natural language and mathematical equations for the documentation. As soon as a concrete tool chain is established in terms of automated EA management metric management method support, we strongly recommend the usage of expressions in formal (programming) languages. As we observed, the definition and agreement of this description element can be considered highly political and emotional in industry. Thus, we strongly recommend to interested organizations to ensure a collaborative definition process of calculation rules and to ensure the agreement and the acceptance of these definitions by all stakeholders involved in the process.

5. The element **Source** is concerned with the documentation of the origin of a metric. As we experienced during cooperations with industry partners (cf. Section 3.5), the majority of the EA management metrics we observed were motivated by suggestion from related (IT) management literature e.g. Cobit [IT09], or ITIL [Of00]. However, in some cases, we observed also the application of the *goal-question-metric* (GQM) approach by Basili et al. [BCR94]. From our perspective, the documentation of a metric source provides an additional benefit towards benchmarking. For instance, if two organizations plan to perform a benchmark on specific aspects of their enterprise architectures or EA management functions, and these organizations are implementing the Cobit framework, the timely identification of concrete metrics for the benchmark can benefit from the information documented by this description element.

6. The element **Layers** is concerned with the documentation of EA layers and cross-cutting aspects affected by the usage of a given EA management metric. Thereby, a metric requires specific EA data. Thus, the documentation of affected EA layers and cross-cutting aspects can enhance a better and more transparent overview and understanding of the quantitatively assessed parts of the EA. In addition, the information provided by this element supports the timely retrieval (search) of metrics.

7. The element **Information model** is concerned with the documentation of the underlying information model (system's structure) of a metric. As described in Section 3.2, we use *UML* class diagrams for the documentation of information models in this thesis. Nevertheless, other modeling languages could be employed as well, e.g. i^*, causal loop diagrams, or the *business process modeling notation* (BPMN) [Ob11a]. This description element fulfills two critical purposes at the same time. Firstly, the information model is aligned with the computation rule of a given metric. Thereby, the model must be minimal in terms of the number of classes, class attributes and relationships between these classes to support the computation rule. Secondly, the model represents the used terms by the metric. In our understanding, every class name, attribute name and relationship name must be in line with the terminology of the involved stakeholders and must be clearly described (glossary).

8. We propose the element **Code** in terms of a unique identifier for a metric. Hence, this element supports the timely retrieval of metrics and can be unambiguously used by the involved stakeholders.

For the category *organization-specific structure elements* we propose the following two metric description elements:

1. The element **Mapping** is concerned with the organization-specific adaption of a metric accounting for the given organizational context. Thereby, three distinct aspects of each model element (class, class attribute, and relationship) need to be mapped to the given organizational context. Firstly, as described above, the information model must be in line with the terminology of the given organization. In experience, every organization has its unique terminology for specific EA elements. Thus, for every model element, a pair of a so-called **Name in model** element and a **Mapped name** (to a corresponding concept in the given organization) is to be documented. In this way, organizations can ensure the adoption of a metric tailored to their terminology and to the understandings of the involved stakeholders. Secondly, for every mapped element, a concrete **Data source**, where the respective data is stored, has to be documented. Hence, organizations can document all data sources required for the metric calculation and link these data sources to concrete information model elements. Finally, for each data source, a corresponding **Data contact** stakeholder is to be defined. This stakeholder is accountable for the provision and quality management of the data required for the calculation of a given metric.

2. The element **Properties** is concerned with the documentation of eight organization-specific metric properties. Firstly, we propose the documentations of an **Owner** and a **Responsible** person (cf. Section 3.2). Then, we recommend the documentation of the intended **Measurement frequency** of a metric, e.g. monthly or quarterly. Further, we suggest to document an **Interpretation** of metric values. In our experience (and in analogy to the element calculation rule), the documentation of this description element can become emotional in industry, since the responsible persons usually fear disadvantages, or negative consequences for not reaching specific predefined goal values. Thus, we recommend to interested organizations to ensure a collaborative definition and documentation process of this element (including at least the owner and responsible persons). Both roles should agree and accept on the documented metric interpretation. Further, we suggest to document a **Target value** and optionally several **Planned values** for concrete points in time. We recommend to perform this task as an collaborative process in analogy to the documentation of the interpretation element and to ensure, that the owner and responsible person agree with the documented values. Further, we recommend to explicitly document tolerance values for a metric. Moreover, we recommend to document organization-specific **Escalation rules** for a given metric. This element is concerned with the documentation of organization-specific rules providing guidance for the responsible person how to react in specific situations, e.g. when to inform the metric owner that a specific metric value is reached or when to request support by the owner, if specific organizational empowerment is missing to implement a countermeasure towards the achievement of predefined metric values. Further, we suggest the documentation of two values for each of the presented properties elements. Firstly, we suggest to document the concrete organization-specific metric *Property value*s. Secondly, we suggest to

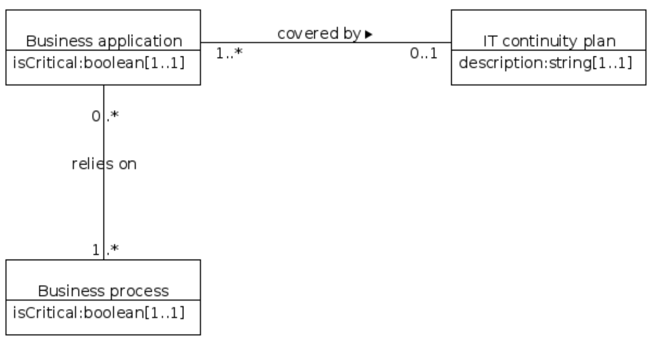
document *Observed values*, if information for the organization-specific values is available from other organizations or from related sources. For instance, for a given metric from related literature a recommendation towards the measurement frequency, e.g. quarterly, can be documented by this element. However, the property value of the metric for the given organization can be set to monthly (cf. Figure 3.23).

IT continuity plans for critical business applications supporting critical business processes

Management summary

All business applications supporting a critical business process must be classified as critical and must have assigned IT continuity plans. The metric supports the monitoring of the implementation progress towards the implementation of this regulatory compliance requirement.

Information model



Goals

Ensure compliance
Foster innovation
Improve capability provision
Improve project execution
Increase disaster tolerance
Increase homogeneity
Increase management satisfaction
Increase transparency
Reduce operating cost
Reduce security breaches

Organization-specific instantiation

Mapping:

Name in model	Mapped name	Data contact	Data source
Business application	Application	M. Maier	EA rep.
isCritical	critical	M. Maier	EA rep.
covered by	covers	C. Schmidt	Risk mgmt. rep.
IT continuity plan	Disaster plan	C. Schmidt	Risk mgmt. rep.
description	description	C. Schmidt	Risk mgmt. rep.
Business process	Business Process	M. Maier	EA rep.
isCritical	critical	M. Maier	EA rep.
relies on	supports	M. Maier	EA rep.

Properties:

Metric property	Property value	Observed values
Measurement frequency	monthly	Quarterly
Interpretation	Good, if = 100% Normal if > 90% Bad <= 90%	
Owner	J. Smith (CIO)	CIO
Responsible	J. Doe	
Target value	100% on 31.03.2015	100%
Planned value(s)	25% at 31.12.2014 65% at 31.06.2014 90% at 31.12.2014	
Tolerance value(s)	3% until 31.12.2014, 0% from 01.01.2015	0%
Escalation rule	none	

Calculation rule

Number of business critical processes relying on critical business applications covered by an IT continuity plan, divided by total number of business-critical processes.

Code

EAM-KPI-0046

Source

Cobit

Layers

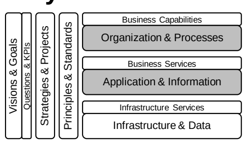


Figure 3.23.: MMFS application for the documentation of an EA metric

The presented MMFS structure allows a uniform documentation of metrics. Thereby, the term uniform refers to an identical representation of metrics. Moreover, the MMFS structure supports organizations in adapting metrics with respect to the given organizational context.

Accounting for our idea to collaboratively define metrics with all involved stakeholders, we identified the need of a suitable graphical representation of the MMFS. During discussions with industry experts, we realized that a representation (template), fitting on a DIN A4 page (cf. Figure 3.23), positively guides the metric documentation process. Thereby, this template can be initially used in a printed paper form to abstract from the metric tooling and to focus the collaboration efforts on the documentation of a metric.

With respect to the three previous subsections, some of the MMFS elements are recommended by literature and some are proposed based on our experience and understanding. Table 3.14 illustrates which of the elements originate from literature and which are unique:

MMFS element type	**MMFS element**	**Kütz [Kü10]**	**Parmenter [Pa10]**	**Popova et al. [PS10]**	**Neely et al. [NAK02]**	**Addicks [Ad10]**	**Stutz [St09b]**	**Strecker et al. [St12]**
General structure elements	Title	✓		✓	✓	✓	✓	✓
	Management summary	✓	✓			✓	✓	✓
	Goals							✓
	Calculation	✓		✓	✓	✓	✓	✓
	Source							
	Layers						✓	✓
	Information Model							
	Code					✓	✓	
Organization-specific structure elements	Name in model							
	Mapped name							
	Data contact	✓						
	Data source	✓	✓	✓	✓	✓		✓
	Owner	✓			✓			✓
	Responsible	✓	✓	✓	✓		✓	✓
	Measurement frequency	✓	✓	✓	✓		✓	✓
	Interpretation							
	Target value	✓	✓	✓	✓		✓	✓
	Planned value(s)	✓	✓		✓			✓
	Tolerance value(s)	✓						
	Escalation rule(s)	✓						

Table 3.14.: Mapping of MMFS elements to elements suggested by literature

The mapping presented in Table 3.14 allows the following five conclusions regarding the MMFS elements:

1. Surprisingly, none of the presented MMFS elements is suggested by all authors. Nevertheless, eight MMFS elements are recommended by literature from all three management areas and thus, should be considered as indispensable parts of a generic metric description structure. Thereby, the elements *Title*, *Calculation*, *Responsible*, *Data source*, and *Measurement frequency* are suggested by 6/7 of the sources. Further, the elements *Management summary*, and *Target value* are suggested by 5/7 sources. The element *Owner* is proposed by one author from each management field.

2. The element *Planned values* is proposed by 4/7 sources (from IT and from management literature). Surprisingly, none of the relevant EA management sources accounts for this aspect of a metric. However, since planned EA states play an important role in the domain of EA management(cf. Buckl [Bu11]), we recommend to document concrete planned metric values next to the target metric values to support the quantitative assessment of planned EA states.

3. The elements *Layers* and *Code* are proposed only by EA management sources (2/3). Therefore, we consider the documentation of these metric description elements as important part of a generic and comprehensive EA management metrics description structure.

4. Each of the following four elements - *Goals*, *Data contact*, *Tolerance value(s)*, and *Escalation rule(s)* - is suggested by only one of the investigated sources.

   Firstly, the documentation of the element *Goals* is proposed only by Strecker et al. [St12]. In our understanding, this element is concerned with one of the most important aspects regarding the definition of metrics. According to Basili et al. [BCR94], every metric must be linked to at least one goal. Therefore, we consider the missing documentation of goals related to a given metric in the corresponding 6/7 metric description structures as a critical limitation of the proposed structures. Secondly the three elements *Tolerance value(s)*, *Escalation rule(s)*, and *Data contact* are suggested only by Kütz [Kü10], i.e., from the field of IT management literature. In our understanding, metric values usually do not exactly match predefined planned and target values. Hence, the documentation of acceptable deviations should be documented to reduce communication efforts in the metric management process. In addition, the definition of concrete *Escalation rule(s)* in collaboration between the owner and the responsible person, can address specific empowerment issues, which occurrence seems very certain. In this way, a timely management reaction can be ensured as a response to the occurrence of such issues. Furthermore, the identification, collection, and automated provision of required data can be considered as one of the most critical activities in a metric management process (cf. Kütz [Kü10], Neely et al. [NAK02]). Hence, we consider the definition of a data contact as a critical part of the documentation process of an EA management metric.

5. The five elements *Source*, *Information model*, *Name in model*, *Mapped name*, and *Interpretation* as proposed by us, are unique (not suggested by the related work). These elements reflect our experience in the domain of EA management with respect to tailoring general EA management concepts to the context of specific user organization. Additionally, none of the related sources distinguishes between *general structure elements* and *organization-specific structure elements*.

As supported by the evaluation results of the MMFS (cf. Subsection 3.4.5), all MMFS elements were evaluated as appropriate and useful. Additionally, the MMFS as a whole was evaluated as complete in terms of a generic and comprehensive metric documentation for the EA management domain.

During the design process of our MMFS structure, we account for a minimal number of proposed elements for a generic an comprehensive metric description. Hence, we do not include all of the proposed metric description elements from literature in our solution. In the following paragraphs we provide a justification for our decisions. Nevertheless, our software prototype supports the organization-specific extension of the MMFS. In this way, organizations can extend our MMFS structure by each of the not included metric description elements, if required (cf. Chapter 5). For the purpose of the decisions justification, we distinguish three reasons to not include specific elements in our MMFS.

Firstly, our structure accounts for the description and documentation of metrics, whereas the management of the metric life-cycle (usage of a metric) should be accounted by an EA management metric management method and its tool support. Thus, we do not incorporate any of the four elements from the category *Presentation* as proposed by Kütz [Kü10] in our MMFS. In analogy to the fields of software engineering and databases, we stick to the principle of separating data (queries) and views. In our understanding, the management of stakeholder-specific views (representations) of a metric should be accounted for by the tool that supports the metric management process. In addition, the concrete types of supported metric visualizations are determined by the used tool. Further, we do not include the element *relates to*, concerned with the documentation of metric relationships to other metrics, in our MMFS as recommended by Neely et al [NAK02]. In our understanding (cf. Section 5.1.4), the management of metric relationships and their evolution is an indispensable part of a holistic metric management method, however, it is not a part of a generic metric description. Hence, we account for this aspect in our software support, where the metrics relationship management can be performed in a controlled environment (cf. Section 5.3). Further, we consider the element *threshold* (proposed by Popova et al. [PS10]) as relevant for the representation of a metric. Additionally, we consider the element *comparison of metric values* (proposed by Addicks [Ad10]) as relevant for the presentation of metrics. Moreover, we do not incorporate the element *linkage to (critical) success factors*. In our understanding, this element (as proposed by the author) is concerned with the identification and documentation of different risks towards the implementation and management of a metric. These aspects should be however accounted for by a holistic metric management method and not by the documentation of a single metric. The same applies for the element *related success factors* as proposed by Stutz [St09b]. Finally, with respect to Popova et al. [PS10], we do not include the element *scale* in our MMFS. The scale of a given metric can be derived from the calculation rule and is a relevant aspect for the representation and interpretation of a metric. Hence, the scale can be documented within the interpretation element and does not require a designated MMFS element.

Secondly, we did not include the subsequent elements, since we consider the documentation of these aspects as redundant for our MMFS.

- With respect to Kütz [Kü10], we decide to not include the element *validity period*. Our MMFS supports the documentation of a measurement frequency, as well as target, planned and tolerance values. Thus, the validity period of a metric can be derived from the MMFS. If the target date is reached for a given metric, the involved stakeholders must decide whether they want to extend the usage of the metric by defining new planned and target values, or to stop using the metric. The same applies to the element *time horizon* as proposed by Strecker et al. [St12].

- With respect to Parmenter [Pa10], we consider the element *teams that have chosen to measure it* redundant, since our MMFS accounts for the documentation of an owner and responsible person. Further, the documentation of the *type* of a metric, i.e., distinguishing between EA metrics and performance indicators is redundant, since this information can be easily derived from the calculation rule of a metric. The same applies for the element *type* as proposed by Popova et al. [PS10] and by Addicks [Ad10], as well as to the element *unit of measurement* as proposed by Strecker et al. [St12].

Thirdly, we do not include specific elements in our MMFS, since they are contradictory to our understanding:

- With respect to Kütz [Kü10], we do not include a *responsible person for the calculation* in our MMFS. With respect to the previously described risks and related countermeasures (cf. Section 3.3), organizations should use a dedicated tool support for their metric management process. In Chapter 5.1.4, we present a corresponding software support. Thereby, the calculation of a metric is performed by the software system using formal DSL expressions (queries) as representations of the calculation rules. Hence, a responsible person for the calculation is not required.
- With respect to Parmenter [Pa10] and according to our understanding, the usage of EA management metrics does not require the application of an EA management BSC. This information can be documented only by organizations using a concrete EA management BSC and hence, should not be a part of our generic MMFS (the same applies to the element *assessment perspective* as proposed by Stutz [St09b]). Finally, the element *time zone*, is described ambiguous in our understanding and is interpreted by us as means for the documentation of planned and target values, which are addressed by our MMFS structure. In addition, our software prototype supports the historicization of metric values. Thus, we decided to not incorporate this element in the MMFS.
- With respect to Popova et al. [PS10], we do not incorporate the element *min value, max value*, since it is ambiguous in our understanding. As previously discussed, if a metric result value is from type ratio, its values are constrained by the interval $[0;1]$ - (0% to 100%). Otherwise (the metric type is absolute number), the metric value is constrained by the interval - $[0;\text{number of existing instances}]$. Further, we consider the element *hardness* misleading. According to Niemi [Ni08], specific EA benefits cannot be easily attributed to a given EA management function or cannot be easily quantified. However, if the MMFS elements calculation rule, information model, as well as the organization-specific mapping of a metric are documented, the metric can be calculated. Hence, we consider the differentiation between “hard” and “soft” metrics as not required.
- With respect to Neely et al. [NAK02], we do not incorporate the element *what do they do*. In our understanding, not all possible countermeasures for the achievement of given target or planned values can be defined in advance. Nevertheless, the MMFSs element *Escalation rule(s)* can be used to document recommendations for the responsible person.

- With respect to Addicks [Ad10], we decide to not include the element *estimated effort for data collection* to our MMFS, since this element is misleading in our understanding. The MMFS requires the documentation and organization-specific mapping of an underlying information model for a metric. If the mapping cannot be performed, e.g. a required data source cannot be assigned to an information model element, the information model does not describe the reality in the organization in an appropriate way and thus, the metric should not be used (cf. Section 3.3). In this case, the organization should either not use the metric, or must define a change project to implement a corresponding data source. Thus, the question of estimated data effort should be considered during the change project and not by the MMFS.

As mentioned previously, we consider the proposed ten MMFS elements as the minimal number of required description elements. Subsequently, we provide a justification for this assumption. The following listing describes the consequences, if a specific MMFS element is removed from the structure:

Title Removing this element from the MMFS structure will prevent the involved stakeholders in the metric management method to quickly understand the purpose of a given metric. In addition, a missing title can lead to heterogeneous names and understandings of a given metric.

Management summary Removing the management summary prevents the common understanding of the motivation, expected benefits, and assumptions for the involved stakeholders. This can lead to several of the previously described risks in Section 3.3.

Goals Removing this element prevents the link between a metric and related EA management goals, which is considered as one of the most relevant risk in our domain. Thereby, if the EA management goals change, the involved stakeholders are not able to identify affected metrics, which have to be adjusted or deleted.

Calculation rule Removing this element prevents the common understanding of the measurement rule for a given metric and in addition, the proper implementation of the metric. These issues are considered as risks in our domain.

Source Removing this elements prevents interested stakeholders in identifying the origin of the metric. In this manner, additional information available in the corresponding metric source is hidden. Further, a possible benefit towards the timely identification of suitable metrics for benchmarks is omitted.

Layers Removing this element prevents the link between a metric and the related EA layers and cross-cutting functions. Hence, the involved stakeholders are not able to relate the metric to their EA structure, which we consider as significant limitation of a comprehensive EA metric documentation.

Information model Removing this element prevents the formal consideration of the relevant EA elements for the metric. Further, this missing element disables the capability of the MMFS structure to support the organization-specific mapping of terminology as well as required data sources and related data contacts, which are known risks in our domain, cf. Section 3.3.

Code Removing this element prevents the timely retrieval of metrics, as well as the establishment of short and comprehensive aliases of metrics by the involved stakeholders. According to our evaluation of the MMFS elements (cf. Section 3.4.5), practitioners consider this element as helpful and important part of a holistic metric description.

Mapping Removing this element prevents the organization-specific mapping of terminology, as well as required data sources, and related data contacts. These issues are known risks in our domain.

Properties Removing this element prevents the documentation of several viral metric management process elements, e.g. target value, measurement frequency, owner, and responsible person for a given metric. This leads to several known risks in our domain.

In the subsequent section we provide an evaluation of the appropriateness of the proposed element and the usability and completeness of the MMFS structure as a whole based on an the results from an online survey conducted with 29 experts from the readers group of our EA metric management catalog (cf. Section 3.5).

3.4.5. MMFS Evaluation

The basis of the MMFS designs evaluation was made by the documentation of 52 metrics observed in German industry in the course of a research project. The main goal of this project was to document EA management metrics used in industry. We firstly collaborated with the IT strategists of a financial services department of large German engineering organization concerned with the uniform and comprehensive documentation of organization-specific EA management metrics. Secondly, we cooperated with one of the enterprise architects of a mid-sized German public bank also interested in the application of our MMFS for the documentation of their organization-specific EA management metrics. In both organizations, a set of metrics was already defined by the experts with the support of different consultancies. Nevertheless, both organizations required a more uniform description of their metrics and hence, they decided to apply our MMFS.

In the course of the project, the predefined mere textual metric documentations have been mapped to our MMFS description structure. Afterwards, all information not included in the original textual documentations, but required by the structure, have been complemented. Very often, this was the case for the *Information model*, *EA management goals*, and the mapping tables containing the organization-specific details. The outcome of the project is documented by our EA metric management catalog [Ma12a] (cf. Section 3.5). This catalog demonstrates the theoretical applicability of the proposed structure and provides a facility to evaluate the artifact's usability with industry domain experts.

We started the assessment of the MMFS's applicability by a two hours' group discussion with six enterprise architects from a large German car manufacturing company in December 2011. As a result, the experts approved the general idea behind our concept. Furthermore, they suggested the renaming of some elements, i.e., "data contacts" instead of "data contact" and "data sources" instead of "data source" allowing for the documentation of multiple entries (since different subsidiaries of the organization should use the same metrics). In our understanding, the MMFS should be however instantiated multiple times - one instance for each subsidiary. In addition, the interviewed experts came up with several ideas how the structure can be extended, e.g. to use (mathematical) formulas or programming language queries for the documentation of the calculation rules next to a description using natural language. In our understanding, this redundant representation can result in inconsistencies, thus we recommend, to use only formal descriptions if possible, otherwise comprehensive descriptions of the calculation rules, but not both types at the same time.

After incorporating the feedback in our MMFS and after publishing the catalog on the web pages of our chair [2] on January 15^{th}, 2012, an online expert survey was conducted to evaluate the relevance of the artifact on an elemental level. The survey took place between April 10^{th}, 2012 and May 21^{th}, 2012 and contained 35 questions. Of this set, 24 were closed questions using a strict five-point Likert scale. Targeting at an academic as well as practitioner audience, we estimated a survey completion time of about 17min. To ensure familiarity with the MMFS and its elements, only those experts were eligible to participate in the survey who had previously downloaded our catalog [Ma12a] (to the given point in time, we had 75 registered readers).

In total the survey has been completed by 27 experts working in seven different European countries - *Germany* (15), *Sweden* (6), *Portugal* (2), *Austria* (1), *Greece* (1), *Denmark* (1) and *Poland* (1). At the time we collected the data, the experts were employed in the industry branches *consulting* (9), *finance* (7), *manufacturing* (3), *education* (2), *telecommunication* (2), *IT services* (2), *energy* (1), and *government* (1). Asked for their professional occupation, following results were collected - *enterprise architect* (13), *consultants* (6), *business architect* (2), *IT architect* (2), *managing position* (2), and *academic and educational occupation* (2). Asked for their relevant working experience in the domain of EA management, the participants provided following feedback - *more than 10 years* (3), *6-10 years* (10), *1-5 years* (12), and *less than 1 year* (3). With respect to the question for their relevant working experience with metrics in management activities, following results were collected - *more than 10 years* (1), *6-10 years* (5), *1-5 years* (12), and *less than 1 year* (9). These results were complemented by the feedback of the participants for following two questions - *How many metrics have you designed?* and *How many metrics are you currently using / monitoring?*, as depicted in Table 3.16.

Please note, that only 23 of the 27 participants answered these optional survey questions. According to the collected feedback, in average, the experts have designed 23 metrics and were in monitoring 5 metrics at the time the survey took place (median values - 10 and 2). Within the survey, each single MMFS element has been firstly evaluated with respect to its appropriateness as a part of a generic, uniform, and comprehensive structure for the documentation of metrics in the EA management domain by the experts. Table 3.18 provides an overview of the collected results.

As the collected results confirm, the majority of the experts (strongly) agreed with the proposed MMFS elements. Nevertheless, one expert strongly disagreed with the documentation of a *Source*, providing the comment “This element does not provide any value”. In our understanding, and in line with the 19 experts who (strongly) agreed with this MMFS element, the documentation of the origin of a metric should be taken into account, since this documentation supports the comprehensive motivation understanding of a metric and provides additional benefits towards benchmarking (cf. Section 3.4). Further, one of the experts strongly disagreed with the element *Information model*, however, no explanation for this rating was provided. Documenting the underlying (structure) model is an essential part of the description of a

[2] `http://wwwmatthes.in.tum.de/wikis/sebis/eam-kpi-catalog`

Responses	How many metrics have you designed?	How many metrics are you currently using/monitoring?
Participant 1	0	0
Participant 2	10	3
Participant 3	5	0
Participant 4	3	6
Participant 5	100	0
Participant 6	0	0
Participant 7	3	3
Participant 8	20	15
Participant 9	20	20
Participant 10	12	12
Participant 11	2	0
Participant 12	5	8
Participant 13	0	0
Participant 14	250	15
Participant 15	15	5
Participant 16	0	2
Participant 17	5	0
Participant 18	0	0
Participant 19	20	20
Participant 20	20	0
Participant 21	15	0
Participant 22	10	0
Participant 23	10	8

Table 3.16.: Participants metric design experience and metric usage

metric. Additionally, with respect to the previously presented risks in Section 3.3, the documentation of an information model mitigates several of the presented risks (e.g. ensuring that a metric correctly represents the reality in an organization or accounting for the definition of standard terms). Thus, and also in line with 15 of experts who (strongly) agreed with the proposed element, we decided to keep this element as part of our MMFS.

Further, some respondents reported that certain EA management goals used in the MMFS are “too general” or “vague” and that “business goals are missing”. Moreover, suggestions for additional goals - “improve resource utilization”, “increase standardization”, and “increase time-to-market” were given and one expert even called for a free-text field in the MMFS allowing the documentation of newly evolved goals. Furthermore, one participant pointed out that planned and target values might be understood as “self-fulfilling prophecies”, meaning that calculation rules and underlying data are purposefully adjusted in order to meet these targets. Lastly, one of the survey respondents recommended the inclusion of an additional element - a cost field providing details on the implementation and management of a metric.

Asked about possible alternatives for the application of UML for the description of a metrics information model, three participants provided improvement suggestion. One participant

MMFS element	Strongly agree	Agree	Neutral	Disagree	Strongly disagree	No answer
Title	18	4	0	0	0	5
Management summary	20	2	0	0	0	5
EA management goals	7	11	3	1	0	5
Calculation rule	13	8	1	0	0	5
Code	12	7	3	0	0	5
Layers	3	12	7	0	0	5
Information model	7	8	5	1	1	5
Source	5	11	4	1	1	5
Mapping	4	9	8	1	0	5
Properties	8	9	3	1	0	6

Table 3.18.: Results from the appropriateness evaluation of the MMFS elements

(from academia) suggested to use *i** [3] models instead of UML models. Two participants (one from services and one from energy) suggested to use BPMN [Ob11a] models. One further participant stated "the usage of UML depends on the maturity level of EA management / SW development in your company. Not many understand the modeling techniques and how to apply outside SW development". This feedback illustrates, that some experts doubt, that UML is the best language choice for communicating information model details to stakeholders from business domains in their organizations. However, in our understanding, BPMN is not appropriate for the description of a system's structure or behavior. In addition, we doubt, that i* supports the communication process of underlying models to managers in a better way, since this modeling language is developed in analogy to UML for usage in the IS domain. As more appropriate alternatives, we think that causal loop diagrams, or stock and flow diagrams might be also understood by business stakeholders, since these types of models have gained already certain interest in different management fields (cf. Forrester [Fo94]). Nevertheless, UML is widely accepted as a modeling language in our domain (cf. Schweda [Sc11]), thus we decide to stick to the usage of UML for documenting information models in our MMFS.

Beside the relevance of each element, the added value of the MMFS has also been evaluated as a whole. 15 out of 18 respondents who answered the corresponding questions, (strongly) agreed with the statement that the MMFS is helpful for the purpose of communicating organization-specific metrics in their organizations with stakeholders from different backgrounds. In addition, 10 out of 18 respondents (strongly) agreed with the statement that the MMFS could become the standard form for describing EA management metrics in their organizations.

In summary, the collected results confirm that the proposed MMFS elements are useful and appropriate for the generic and comprehensive documentation of metrics in the EA management domain. While this result is not unexpected for the general description elements (since

[3] `http://www.cs.toronto.edu/km/istar/`

the majority of these elements is also described in many of the related sources), it is in particular surprising, that the proposed organization-specific MMFS elements, which are either unique or very rarely discussed in literature, were also (strongly) agreed by the experts.

3.5. Metric Management Catalog

As reported initially by Kaisler et al. [KAV05] in 2005 and as confirmed by Lucke et al. [LKL10] in 2010, the current EA management literature does not provide recommendations for concrete metrics related to specific EA management problems. This holds in particular true, for our pool of related sources in the EA management domain (cf. Table 2.2). These sources provide only few metric examples, however structured collections or practice proven metrics for specific EA management problems are not provided. Surprisingly, even the TOGAF framework [Th09], which we consider as one of the most popular and widely-accepted EA management frameworks in industry (since the framework is used as a foundation for several professional certification programs), only states that metrics (in the terminology of the authors *KPI*) must be used to provide quantitative information for a given EA management function. Moreover, TOGAF does not provide any guidelines how to develop or how to document metrics, and even more does not give a single metric example. Further, one of the popular IT management frameworks in industry - Cobit [IT09] - provides a list of several hundred metric suggestions, however, for the domain of EA management these metrics face the following five critical limitations:

1. A focus on EA management goals is missing, since the framework is designed for the application in the domain of IT process management.
2. The presented metrics are described too general by only a short textual statement.
3. No uniform documentation structure for the metrics is used.
4. No details regarding the required computational data is given.
5. No support for the organization-specific adoption and implementation of the metrics is provided.

On the other hand, the demand for quantitative models in the domain of EA management grows in industry as confirmed by several experts. Here, we see several parallels to the situation in the years 2005 - 2008, when practitioners demanded practice proven EA management patterns for specific EA management problems. Back in these days, many organizations started to introduce and establish organization-specific EA management functions. Many of the problems they faced, were concerned with the missing recommendations for concrete qualitative models addressing reoccurring EA management problems. Additionally, several problems were reported with respect to the identification, collection and visualization of related data for qualitative EA models. To provide a sufficient support for the practitioners and to address this problem in academia, our chair started to observe existing solution in industry with the goal to identify and document EA management patterns and anti-patterns for specific problems. In 2008, Buckl et al. [Bu08] published the EAMPC, comprising over 120 EA management patterns and anti-patterns. This catalog received significant attention in industry and was used worldwide by a multitude of organization.

Referred to the current situation with respect to the missing collection and recommendations for concrete metrics in the EA management domain, in the end of the year 2011 we decided to follow the idea of applying a pattern-based problem solutions and started with the observation of metrics used by practitioners (cf. Section 3.4.5). As of December 2011, we successfully documented 52 metrics used in German industry and started to prepare the publication of the results. Thereby, we decided to address two different audiences - practitioners interested in concrete metrics as well as researchers striving for a comprehensive overview on the topic. We consolidated and merged our metrics in a catalog as means of an organized metric collection guiding its users to easily identify recommended best-practice solutions according to their problems. We refer to this catalog as to **Metric Management Catalog**.

At this point in time, we want to highlight one typical misunderstanding regarding the usage of the term quantitative models as we experienced during several talks with experts in industry. Thereby, the practitioners often confounded qualitative models and quantitative models, e.g. the sentence “What programming languages do we employ in our organization?” was frequently given as an example for a metric. According to our understanding, this interpretation is however wrong. The above concept describes a qualitative EA model - a set of programming languages using a nominal scale, e.g. Java, Cobol, or Scala. We use the term *to quantify* according to its dictionary definition - “to express or measure the quantity of something” [4]. In our understanding, the sentences “How many programming languages do we employ in our organization?”, or “What is the amount of employed object oriented programming languages in our organization with respect to all employed programming languages” are valid examples for quantitative models. Therefore, the result type of a metric is always a numerical value - either an absolute number or a ratio. Thus, we documented only those metrics observed in industry, which fulfill our understanding of a quantitative EA model.

To ensure a proper navigation support for catalog readers regarding the timely identification of relevant metrics, we designed and implemented a twofold navigation structure. As presented by Buckl [Bu11] and by Schweda [Sc11] in the BEAMS framework (cf. Section 2.1.3), a problem can be represented as a pair of a corresponding goal and concern. Thus, our catalog firstly supports the navigation and retrieval of metrics based on the EA management goals related to a given metric. We refer to this navigation aid as to **Goal-Metric-Matrix**. The implementation of this navigation option strongly benefited from the element *Goals* of our MMFS structure. Table 3.20 provides an excerpt of this matrix as published in our catalog [Ma12a]. Thereby, every column of the matrix represents an EA management goal, and every row a metric from the catalog encoded by its unique code and pointing to the page number of the corresponding MMFS documentation of the metric in the catalog. Further, if a metric relates to a given EA management goal, the corresponding cell in the matrix is marked with a check mark (✓) character.

Secondly, our catalog supports the navigation and retrieval of metrics based on the related concerns (EA layers and cross-cutting aspects). We refer to this navigation aid as to **Goal-Concern-List**. The implementation of this navigation aid benefited strongly from the element *Concerns* of our MMFS. The catalog provides a textual list of all metrics related to specific elements of our overall EA structure (cf. Section 2.1.1). Figure 3.24 presents an excerpt of this list for the EA layer *Application & Information*.

[4] `http://www.oxforddictionaries.com/definition/english/quantify`

	Ensure compliance	Foster innovation	Improve capability provision	Improve project execution	Increase disaster tolerance	Increase homogeneity	Increase management satisfaction	Increase transparency	Reduce operating cost	Reduce security breaches
EAM-KPI-0001 (p.19)			✓		✓					
EAM-KPI-0002 (p.20)				✓	✓					
EAM-KPI-0003 (p.21)	✓							✓		

Table 3.20.: Excerpt of the Goal-Metric-Matrix according to our catalog [Ma12a]

Application & Information

- EAM-KPI-0001: Application continuity plan availability (p.19)
- EAM-KPI-0004: Costs of inadequate change specifications (p.22)
- EAM-KPI-0012: Application criticality ratings (p.30)
- EAM-KPI-0031: Application portfolio methodology analysis (p.50)
- EAM-KPI-0033: Business application technology standards compliance (p.52)
- EAM-KPI-0039: Defects uncovered prior to production (p.58)
- EAM-KPI-0043: Business applications compliant with IT architecture and technology standards (p.62)
- EAM-KPI-0045: Service desk calls cased by inadequate training (p.64)
- EAM-KPI-0046: IT continuity plans for critical business applications supporting critical business processes (p.66)
- EAM-KPI-0049: Reopened incidents (p.69)

Figure 3.24.: Excerpt of the Goal-Concern-List according to our catalog [Ma12a]

Furthermore, we created an integrated information model by merging the 52 information models of our metrics as illustrated in Figure 3.25. In addition, we mapped the classes of the models to the corresponding EA layers and cross-cutting functions (concerns). Surprisingly, this representation of the metric collection was considered as helpful navigation aid by several industry experts, we talked to during presentations of our catalog. Many of them stated that looking at concrete classes and having an overview on the affected layers at the same time is a helpful mechanism to identify EA elements, which should be quantified in their organizations. According to Figure 3.25, the classes *IT project*, *Business application*, and *Employee* have the most relationships. Hence, we hypothesize that in the beginning of EA initiatives these three classes seem to be the most important concepts which are subject of quantification. Nevertheless, the integrated information model covers all EA layers and cross-cutting aspects,

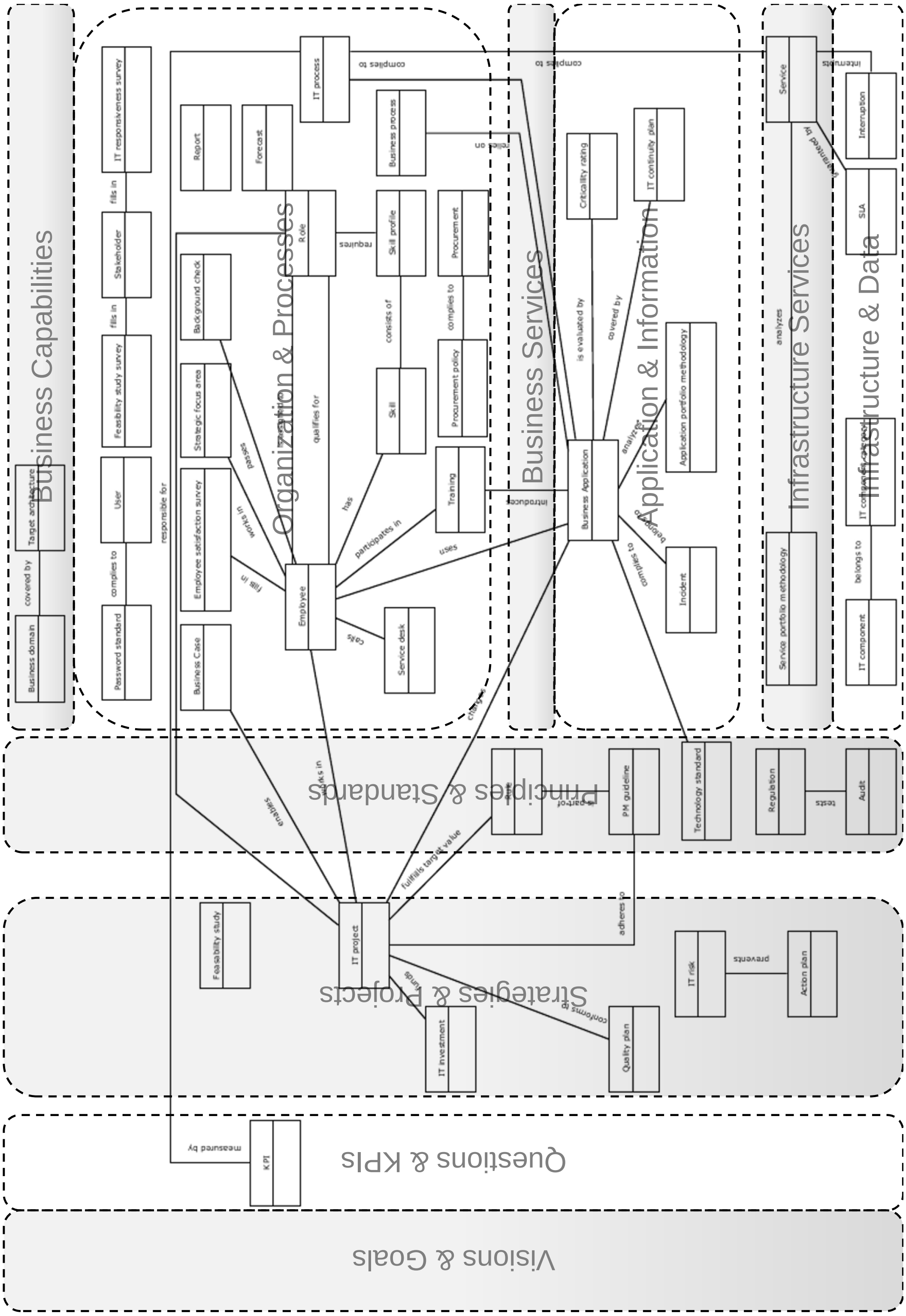


Figure 3.25.: Integrated information model based on the 52 catalog metrics mapped to EA layers and cross-cutting functions according to [Ma12a], cf. Figure 2.1

hence, we can confirm, that the entire overall EA structure is a subject of quantification in industry.

To illustrate the applicability and the navigation support offered to the catalog readers, we provide the subsequent example:

Example 3.3: Search for recommended metrics by EA management goals. An enterprise architect is concerned with the identification and adoption of best-practice metrics related to the EA management goal *Ensure compliance* and the concern *for applications*, which we consider as a part of the EA layer *Application & Information.* Examining the corresponding column in the Goal-Metric-Matrix, the reader discovers a set of 20 related metrics. Additionally, by examining the Goal-Concern-List, the enterprise architect discovers a set of 9 related metrics. The intersection of both sets (metrics related to the given EA management goal and concern) consists of five metrics. The titles of these metrics are: *Application criticality ratings*, *Application portfolio methodology analysis*, *Business application technology standards compliance*, *Business applications compliant with IT architecture and technology standards*, and *IT continuity plans for critical business applications supporting critical business processes.* Equipped with this information, the enterprise architect studies these metrics in detail by reading the corresponding MMFS documentations and decides which of these metrics can and should be adopted by his organization

Further, as required in Section 3.4.4, we accounted for the minimality of the 52 information models documented in our catalog. Therefore, in a pair modeling activity with my college Dr. Christopher Schulz, we defined 52 UML models based on the calculation rules of the metrics. Then, for each of these 52 information models, we tested, if the calculation rule can be still computed after removing a single model element (a concept, an attribute, or a relationship). After finishing this modeling activity, we asked our colleague Alexander W. Schneider, to independently perform the minimality test as described. Hence, after incorporating his feedback, we can assume that all 52 models fulfill our information model minimality requirement.

We published our catalog on 15^{th} of January, 2012 at the pages of our chair. Thereby, accounting for the terminology of the majority of the practitioners, we talked to during the course of our metric identification and documentation research project, we decided to publish the document under the name *EAM KPI Catalog v. 1.0* instead of *EAM Metric Catalog v. 1.0.* In addition, we announced the publication of the catalog on our web pages, as well as on our news feeds which are subscribed by both—practitioners and researches. Moreover, we promoted the catalog also at the *EAMKON* and *Softwareforen Leipzig* practitioner conferences during the year 2012 to spark interest in German industry.

As of April 2^{nd}, 2014, 347 persons have registered and downloaded a PDF copy of the catalog. We consider this fact as an indicator for the relevance of the catalog. By examining the email addresses of the registered catalog readers (accounting only for company emails and ignoring private e-mail addresses hosted for example by Google, Yahoo, etc.) we found out that the readers are distributed all around the globe in the following 22 countries (sorted alphabetically) - *Australia*, *Austria*, *Belgium*, *Bulgaria*, *Canada*, *Denmark*, *Finland*, *France*, *Germany*, *Italy*, *Norway*, *Poland*, *Portugal*, *Qatar*, *Saudi Arabia*, *South Africa*, *Sweden*, *Switzerland*, *The Czech republic*, *The Netherlands*, *The United States of America*, and *The United Kingdom.* Although, we want to highlight also the fact, that (based on our e-mail address analysis) we do not have registered readers from several countries with largest economies, e.g. *Brazil*, *China*, *India*, *Japan*, and *Russia* to the given point in time. To ensure the anonymity of the registered readers, we do not provide any concrete company names, however, we confirm, that following industry branches are represented by the readers - *aerospace*, *chemistry*, *consulting*, *education*, *energy*, *engineering*, *finance*, *government*, *healthcare*, *IT products & tools*, *IT services*, *manufacturing*, *telecommunication*, and *transportation & logistics.*

As the number of the registered catalog readers, their distribution around the globe, and the represented industry branches confirm, there is a clear demand for concrete quantitative models related to specific problems in the EA management domain. We are convinced that further metrics will emerge over the time in industry and that these metrics must be documented in our catalog as well. Further, additional knowledge about metric anti-patterns in terms of metrics, which proved to not be suitable for specific EA management problems, must be a subject of documentation as well. Nevertheless, our catalog (in terms of organized metric collection) provides the first step towards an uniform metric documentation in the domain and someday, it might even serve as a pool of standard metrics used for benchmarking EA management initiatives. Thus, we encourage practitioners to continue sharing metrics and experiences with us, to enable a continuous evolution of the catalogs metrics.

3.6. Metric Management Method

With respect to our research question 5, in this section we present a holistic metric life-cycle management method for the EA management domain. Additionally, we present an integration of this method within the BEAMS framework (cf. Section 2.1.3). The reasons for the integration of our method in the BEAMS framework are:

1. BEAMS supports the definition and the managed evolution of organization-specific EA management functions.
2. BEAMS is based on the application of EA management best-practice solutions (building blocks) for specific EA problems (goals and concerns).

As described in Section 2.1.3, the BEAMS framework is built upon a *method base* representing an organized collection of practice proven building blocks for reoccurring EA management problems. Thereby, these building-blocks are extracted from the EA management patterns presented in the EAMPC [Bu08]. The framework distinguishes in particular between four types of building blocks - information building blocks are concerned with the documentation of qualitative EA information models, visualization building blocks are concerned with the documentation of qualitative EA visualizations, method building blocks are concerned with the documentation of specific EA methods, and glossary building blocks are concerned with the documentation of organization-specific EA terminology. These building blocks are set in relation to each other to design a (qualitative) EA solution for a given problem accounting for the given organizational context. Nevertheless, the framework does not propose a specific type of building blocks to support the design and development of organization-specific quantitative EA solutions. Hence, in a first step, we extend the BEAMS method base by the metrics documented in our catalog (cf. Section 3.5) to close this gap. Thereby, all metrics (documented using our MMFS structure) are added to the method base. Since our MMFS structure accounts for the same EA management goals and concerns as the BEAMS framework (per design), the proposed extension is conform to the underlying BEAMS concepts for the usage and management of its method base. In addition, the two navigation structures offered by our metric catalog are adopted as well. These navigation structures support the identification of metric best-practices from the model base in analogy to the existing BEAMS mechanism for the identification of building blocks relevant for a given EA management problem. Secondly, we propose a holistic life-cycle management method for metrics in the domain of EA management and integration within the BEAMS framework as illustrated by Figure 3.26 and as described subsequently(for the documentation of the method we apply BPMN [Ob11a])

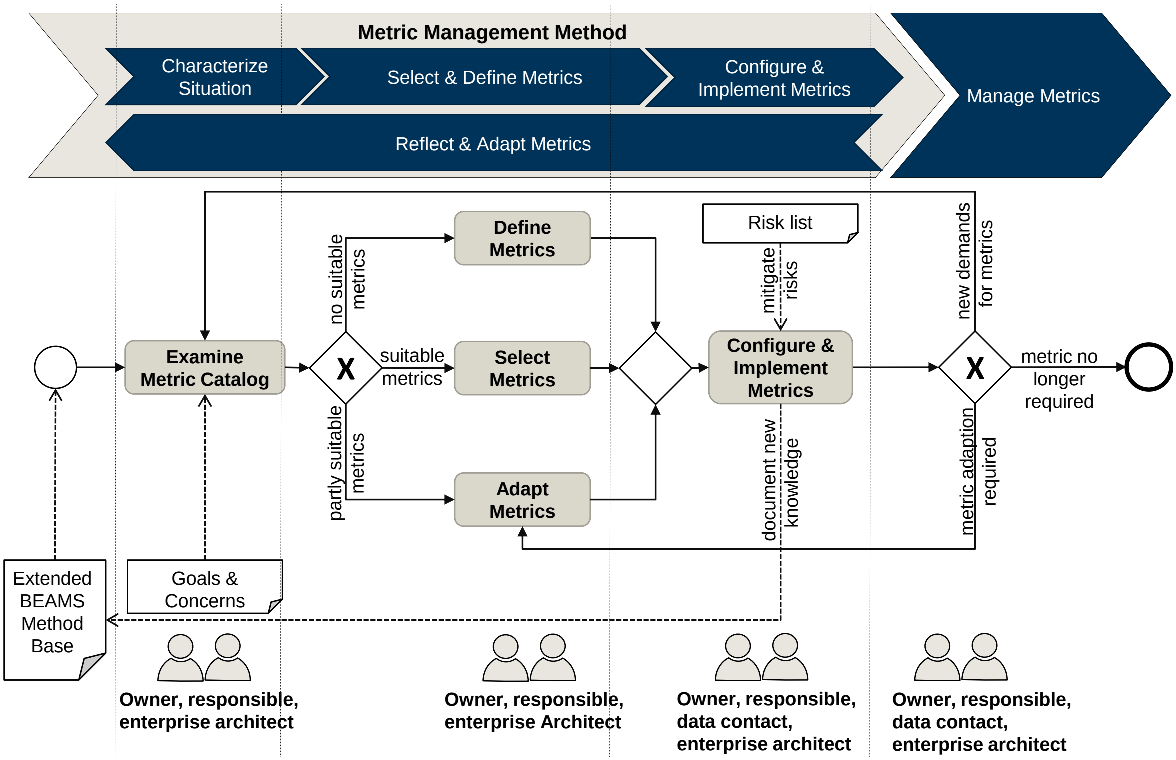


Figure 3.26.: Holistic life-cycle metric management method

Starting with the extended BEAMS method base as described above, the first step of our method (*Characterize Situation*) is concerned with the identification of recommended metric best-practices based on a pair of goal and concern. Thereby, the following stakeholders are involved in this method step - the owner, the responsible person and advisably an enterprise architect. These stakeholders firstly collaboratively document the relevant pairs of goals and concerns. Then, they collaboratively explore the method base to retrieve recommended metrics according to their documented demands. Here, the two navigation structures (the goal-metric-matrix and the concern-metric-list) are used to support the timely retrieval of related metrics from the method base. This method step is performed iteratively for all documented pairs of goals and concerns. The output of this step is a set of recommended metrics. Nevertheless, for specific problems (not covered by the catalog), the result set might not contain recommendations.

The second step of the method - *Select & Define Metrics* - is concerned with the selection of recommended metrics from the first step and with the acceptance of all general structure elements for these metrics by the involved stakeholders. The step is performed iteratively for each metric from the output of the previous method step and for each pair of goal and concern, for which no metric is recommended by the method base. Hence, the following three cases are distinguished and supported by our method:

1. According to the understanding of the involved stakeholders, a recommended metric from the first method step provides a valid quantitative model with respect to the EA of the given organization. In this case, the stakeholders agree with the proposed general structure elements of the metric by executing the sub step *Select Metrics* of the method and additionally, the metric is put in the result set of this step.

2. According to the understanding of the involved stakeholders, a recommended metric provides a partially valid model for their EA, however, the metric can be used after adaption to the given EA by changing specific general description elements. For instance, specific adjustments of the information model (by adding, removing or changing model elements) of the recommended metric might be needed to adapt the metric to the given EA. In this case, the method step *Adapt Metrics* is collaboratively performed by the involved stakeholders. During this step, each of the MMFS general structure elements can be changed, however, the stakeholders have to agree with the proposed changes at the end of this activity. In particular, they have to ensure, that the information model represents a valid excerpt of their EA and that the information model is minimal in terms of supporting the calculation rule.

3. A metric for a given pair of goal and concern is not recommended by the method base or the involved stakeholders disagree with the selection of a recommended metric, since the metric does not fit to the given EA and an adaption is not possible as well. In this case, the stakeholders define and document a new metric by executing the sub step *Define Metrics* of our method. Nevertheless, according to the related literature, the definition of metrics is considered as a complicated and timely consuming task. Thus, we strongly recommend the application of the GQM approach by Basili et al [BCR94] or the application of Stutz's metric definition method [St09b] for this definition process.

The outcome of this second method step is a set of metrics with fully described general description elements ready for the organization-specific configuration and implementation. Thus, in the third step *Configure & Implement Metrics* of our method, the selected metrics from the previous step are iteratively tailored to the given organizational context. For this purpose, all organization-specific MMFS elements (cf. Section 3.4.4) are collaboratively documented by the stakeholders. In contrast to the first two steps, the data contact stakeholders are involved as well in this step. For each metric, firstly the organization-specific mapping of the information model is performed. Thereby, related data sources are identified and the intended data provision (data access) is negotiated with the data contacts. Secondly, the organization-specific properties of each metric are collaboratively defined and documented by the stakeholders. Furthermore, if a multiple instantiation of a metric is required, e.g. to support the adoption of a metric in multiple subsidiaries of the given organization with different responsible persons and different data contacts, all of these stakeholders have to be involved in this step as well. It is essential to ensure the agreement, acceptance, and common understanding of the documented organization-specific description elements within the related stakeholders. For this purpose, for each required instance, a separate metric instance (MMFS description) is created. In our understanding, the general structure elements remain thereby unchanged, whereas the information model mapping and the organization-specific metric properties can differ. For example, accounting for the specific situation in the different subsidiaries, various planned and target values for the metric instances as well as different responsible persons and data contacts can be defined and documented. The result of this third method step is a set of metric instances, where all metric description elements are fully documented and these instances are ready for their implementation. In our understanding, the metric instances must be implemented within the EA management tool of the given organization (in terms of a collaborative and controlled environment for the EA quantitative models). In this way, the organization can ensure that all of the required calculation data as well as all metric instances are stored in the EA repository (single point of truth). Further, a read-only access (or data import into the EA repository) from different enterprise data silos (e.g. project portfolio management repository or risk management repository) is sufficient for the calculation and management of the metric instances (i.e., no changes have to be propagated to the corresponding data sources). Additionally, we propose the implementation of the metric instances as derived attributes within the EA repository (cf. Chapter 5) and the definition, implementation and management of specific access rights for the implemented metric instances within the given EA management tool. In our understanding, only those stakeholders should have (read-only) access to a given metric instance, who are involved in its life-cycle management (i.e., the owner, the responsible person, the data contacts and advisably the involved enterprise architect).

The life-cycle management of the implemented metric instances is addressed by the method step *Manage Metrics* in our method. According to the MMFS documentations, the metric instances are automatically calculated and the results are reported to the involved stakeholders within the EA management tool. Our method accounts for changes in the life-cycle of metric instances by the three activity flows - *new demands for metrics*, *metric adaption required*, and *metric no longer required*. The different life-cycle changes can have the following reasons:

1. The target point in time for a metric is reached. In this case, the stakeholders have to define a new life-cycle of the metric instance. If the target value is reached, and a further usage of the metric is not intended by the owner, the metric instance is to be deleted to ensure that no efforts (in terms of budget) are invested in managing not demanded metrics. If the target value is reached, but the metric owner request a monitoring of the results in terms of ensuring a constant metric value behavior over the time (e.g. service level agreements for the availability of specific business applications), corresponding planned and target values for future points in time are to be defined. For the case that the target value is not reached yet, new planned and target values have to be collaboratively defined or the given metric instance is to be deleted, since no agreement about a new target can be ensured.

2. EA management functions support and guide organizational changes over the time. Hence, EA management functions are subject of change as well. In this case, the set of implemented metrics instances must be revised whenever EA goals and concerns change. Furthermore, the involved stakeholders in the life-cycle management of the metric instances can change over the time as well, e.g. based on restructuring initiatives or personnel changes in their organizations. In this case, all metric instances affected by the change of a given stakeholder, must be revised as well.
3. Over the time, new demands for metrics can arise, and thus, new metric instances are required. Furthermore, the stakeholders gain experience with the metrics they manage. Additionally, both—positive and negative influences based on relationships between metrics can be observed and analyzed during the metric management process. Hence, improvements of implemented metric instances can become necessary.

Therefore, our method supports the previously described three activity flows to address change requests during the metric life-cycle management process.

With respect to Section 3.3, our method accounts for several of the presented risks and recommended countermeasures. Firstly, the usage of our MMFS structure accounts for the following six risks:

- By the documentation of the related goals and concerns, as well as the documentation of the underlying information model, calculation rule, and management summary, the occurrence of the risk *Use of irrelevant metrics* is mitigated.
- According to Section 3.4, all indispensable metric description elements, as suggested by the related literature, are incorporated in the design of our MMFS structure. Additionally, the MMFS can be extended by user organizations to document additional metric description elements according to the understanding and expectations of the involved stakeholders, e.g. BSC perspectives, or uncertainty regarding the existence and actuality of specific EA model data. Thus, the four risks *Metrics are not properly defined*, *Metrics are too abstract*, *Standard terms*, and *Static metric structure* are addressed by our method.
- Based on the collaborative documentation of planned and target values by the involved stakeholders in the MMFS structure, the risk *Short term focus only* is addressed by our method as well.

Secondly, the usage of our organized metric best-practices collection, accounts for the risk *Customer satisfaction is not measured*. The method base contains the two metrics - *Customer satisfaction index* and *Employee satisfaction index*, which are recommended by our method for specific problems.

Thirdly, our method itself accounts for the following twelve risks:

- Based on its collaborative process design, our method ensures the acceptance, agreement, and commitment of all involved stakeholder with respect to the implementation and management of the metric instances tailored to the context of their organizations. Thus, the risks *Resistance to change*, *Unchanged targets*, *Goal displacement*, *The metric presentation causes divergent interpretations*, and *Vague organizational goals* are accounted for by our method.

- Our method further accounts for the automation of the data collection, computation and reporting processes for the implemented metric instances within the used EA management tool in a given organization. Thus, the risks *Missing automation of the measuring process*, *The metrics team is not constantly informed*, and *Defective data* are addressed.

- By the documentation of related data contacts, as well as the involvement of these stakeholders in the method steps *Configure & Implement Metrics* as well as *Manage Metrics*, the following two risks are addressed by our method - *Considering too much (or too less) data*, and *Access to confidential data.*

- The proposed integrated tool support for our method within the EA management tool of user organizations addresses the risks *Manipulated outcomes* and *Ambiguous relationships between metrics* by providing transparency over the EA model data, the relationships between implemented metric instances, as well as by explicitly managing the access rights of the involved stakeholders on metric instance level.

Nevertheless, the following four risks remain uncovered by our method, and hence, during the activity flow *mitigate risks* of the method step *Configure & Implement metrics*, the stakeholders have to ensure that none of these risks occurs for the metric set prepared for implementation:

1. *Using too many metrics* - according to the literature, there is no "magic" number for the size of a metric set for a given EA management function. Thereby, the majority of the authors state, "one should use as much metrics as required and as less as possible". We additionally recommend that every time when the metric set is revised during the *Manage Metrics* step of the method, the stakeholders double check that all of the existing and managed metrics are still required.

2. *Setting extreme target values* - the definition of target and planned values is done according to the given organization context and the understanding of the involved stakeholders. In some cases, the recommended metrics provide concrete best practices shared by other organizations using these metrics. Over the time and during the regular monitoring of the measured values, the stakeholders will gain more knowledge with respect to the predefined target values. If at a given point in time the target value is considered as unrealistic by at least one of the stakeholders, it should be collaboratively adjusted (changed).

3. The two risks *Delaying rewards* and *Attach rewards to metric too soon* are not addressed by our method as well. In our understanding, these aspects significantly depend on the given organizational context and the involved stakeholders. We strongly suggest to not attach rewards to newly defined metrics, since rewards can influence a wrong behavior of the stakeholders (cf. Section 3.3 - the phenomenon of "meeting the numbers" [LDH08]). Thus, we recommend to use rewards only if required by the given organization and after initial experience with the given metrics is gained.

Our method supports a learning mechanism in terms of method base extension accounting for the new knowledge gathered in the application of our method. As illustrated in Figure 3.26, by performing the activity flow *document new knowledge*, the method base is extended by the new knowledge provided by the method step *Configure & Implement Metrics.* Thereby, by executing the two sub steps *Adapt Metrics* and *Define Metrics*, new and adapted (changed) metric best-practices are defined. Hence, the following knowledge is added to the method base by our learning mechanism:

- For every newly defined metric, all of the general metric description elements are directly stored into the method base. In addition, observed organization-specific metric properties (e.g. target value or measurement frequency) can be documented too, if shared by the given organization.
- For every adapted (changed) metric, a new metric best-practice is documented in the method base as described in the previous case. In addition, both metrics are marked as versions of each other. In future, having documented concrete metric versions for a given problem and user organization, one can study these versions with research interest to identify dependencies between the metric versions and specific elements of the organizational context (e.g. one version is used by enterprises with centralized IT organization vs. another version of the metric is used by enterprises with federated IT organization). Hence, over the time, by documenting and analyzing metric best-practice versions, more precise recommendations an be supported based on the consideration of specific organizational context elements.

After each extension of the method base, the two navigation structures (the goal-metric-matrix and the concern-metric-list) are updated as well to support the retrieval of the new best practices knowledge. With respect to the tool support of our method (cf. Section 5), all users (readers) of the method base (other stakeholders of the same organization or other organizations using the method base) can be automatically informed about method base extensions, if they subscribe a corresponding notification service. Moreover, as described in Section 3.5, the user organizations and the involved stakeholders will gain experience over time with their metrics and might observe metric anti-patterns for specific problems. Thus, our learning mechanism supports the reporting of anti-patterns as well, to ensure the continuous improvement and extension of the method-base also by documenting observed metric anti-patterns.

CHAPTER 4

Metric Management Method Evaluation

4.1. Evaluation

To evaluate our metric management method (cf. Section 3.6), we followed a twofold approach. As published by us in [Ma12b], we conducted five confirmatory expert interviews concerned with the design, completeness and appropriateness of the proposed method. The results of these expert interviews are described in the subsequent subsection. Afterwards, we applied our method in a case study with a large German and internationally operating engineering company. The results of this case study are described in Section 4.1.2

4.1.1. Method Evaluation via Expert Interviews

To evaluate our metric management method, we initially carried out a series of confirmatory expert interviews by both—telephone and personal talks as described in Matthes et al. [Ma12b]. The primary objective of these interviews was to obtain qualitative feedback from EA management experts experienced in the usage of metrics in this domain. Furthermore, we intended to capture ideas helping to enhance the artifact in future research.

The interviews were conducted between June 21^{th} and June 29^{th}, 2012. In total, we had the possibility to question five EA management experts (three consultants and two IT architects) from four different organizations. The main criterion for the selection of these five experts was their relevant working experience with metrics in the domain (the average experience of the experts was 6.63 years).

Both—the personal and telephone interviews lasted 45 minutes and were subdivided into two parts. In the first part of each interview, we introduced our method during a 30 minute presentation. In the second part of the interviews, we asked the participants to complete a predefined survey. Taking the recommendations of Frazer et al. [FL00] into account, the survey form was limited to one page and contained 13 concise questions. The survey was in turn subdivided into two main areas: participant's background and questions about our method. For each of the questions regarding the design of our method, the following four answers were offered to the interviewees - *I agree*, *Neutral (Neither agree, nor disagree)*, *I disagree*, and *No answer*. Additionally, for every question a comment field was offered to document additional feedback, thoughts and recommendations of the interviewees.

According to the collected feedback, all experts confirmed that the proposed metric management method supports a holistic life-cycle management of metrics. Additionally, all experts confirmed the benefit provided by consideration of metric best-practices in the first step of the method (*characterize situation*). Further, all experts agreed that the separation between a general metric definition (performed in the method step *select & define metrics*) and its organization-specific configuration (performed in the method step *configure & implement metrics*) is helpful and appropriate. Four experts approved that the method can be embedded in their enterprise context, however one selected the answer option *No answer* with the justification, that his organization is a consultancy, which does not have an own EA management initiative. All five experts emphasized that the proposed metric management method possesses a much higher level of detail than the approach they are currently using / have previously used.

Four of the experts confirmed, that the proposed method supports organizations in making their EA management goals measurable. One expert selected the option *Neutral (Neither agree, nor disagree)*. The expert provided therefore the following justification - up to his understanding, an organization requires more precise goal descriptions than just a related general EA management goal. Thus, he proposed the extension of the MMFS structure by an additional organizations-specific description element to support a more precise documentation of organization-specific EA management goals related to a given metric. Nevertheless, the expert stated "based on the fact that your MMFS structure accounts for the documentation of general EA management goals related to a metric, as well as the documentation of target and planned metric values, I tend more to select the *I agree* option than the *I disagree* one". Further, four of the experts appreciated the roles proposed and supported by our method. One selected the option *Neutral (Neither agree, nor disagree)* and remarked that the term stakeholder is too generic and should be refined. Further, one other expert suggested to rename the role *data contact* into *information steward* which is a more established term to his understanding in the domain of EA management.

The last (open) question of the survey was concerned with capturing additional improvement suggestions for our method. Thereby, one expert suggested to define a maturity level system based on the example of the Cobit [IT09] framework, which can indicate the maturity level required for an organization interested in the adoption of a given metric (this idea is in line with the work of Plessius et al. [PSP12], cf. Section 2.3.4). One further expert highlighted the importance of the method step *manage metrics*. In his understanding, "the definition and instantiation of a metric is the first and easy part of a metric management process". However, the expert reported based on his experience that the regular data collection and continuous reporting of metrics are associated with high efforts and hence, with high costs. Therefore, the expert agreed with the importance, correctness, and completeness of the proposed activity flows for the life-cycle management of metrics by our method. Further, the expert emphasized the importance of a corresponding tool supporting the method.

4.1.2. Application of the Method in Industry

In May 2012, we were contacted by the CIO office of an IT department of a large German engineering company. This IT department is internationally operating and responsible for the IT support of the financial services department of the organization. The CIO office of the IT department requested the support of the EA management team for the definition and the management of a metric system to monitor and guide the improvement of its *service management* initiative to predefined goals. To support the organization in the development of the desired metric system, we proposed to apply our metric management method within a case study. Hence, we conducted a two hours' workshop, in which we presented our metric management method and relevant concepts from the related work (cf. Section 2.3.8). After the presentation, the experts confirmed their interest in the application of our method and agreed to perform the suggested case study as a Master's thesis. Hence, we supervised Jawid Rassa, who participated regularly in workshops with both—employees from the CIO office as well as enterprise architects and described the case study in detail in his Master's thesis [Ra13].

During the initial workshop, the CIO showed interest in the development of a BSC tailored to the needs of his organization. However he disagreed with the usage of the four perspectives presented by Stutz [St09b] with the justification, that these perspectives do not cover all related aspects. Hence, we conducted a literature review concerned with the identification of different BSC perspectives in economics and IT management literature as described by Rassa [Ra13]. Thereby, 31 perspectives proposed by 16 different sources were initially identified and documented during the literature review, however, several of these perspectives describe similar aspects. Thus, during a second workshop with one of the CIO office employees, the documented perspectives were discussed and consolidated. The result of this workshop was a list consisting of 14 perspectives. In a third follow-up workshop, the consolidated perspectives were presented to the CIO and his employees, and the following 7 perspectives were selected for the desired metric system - *financial*, *collaboration*, *employee*, *process*, *project management*, *supplier management*, and *governance, risk and compliance (GRC)*.

Starting with this BSC, in a follow-up workshop with one of the CIO office employees, the predefined IT goals of the organization were mapped to the proposed perspectives as depicted in Table 4.1. Additionally, a mapping of these IT goals to the general EA management goals supported by the BEAMS framework was performed.

Perspective	Goal	Mapped BEAMS goals
Collaboration	Increase customer satisfaction	Increase management satisfaction
	Meet customer expectation	Increase management satisfaction Reduce operating cost
Employee	Increase employee satisfaction	Increase management satisfaction
	Availability of staff capabilities for future IT challenges	Increase management satisfaction Increase capability provision
Project management	Determine innovation rate	Foster innovation
	Project delivery in time, budget and quality	Improve project execution Increase management satisfaction Increase transparency

Perspective	Goal	Mapped BEAMS goals
Financial	Costs reduction through improved productivity	Reduce operating cost
	Improved accuracy of cost forecasts	Increase management satisfaction Increase transparency
	Budget adherence	Increase management satisfaction Increase transparency Reduce operating cost
	Complete charging of the IT costs	Ensure compliance
Process	Tool supported ITIL based IT operating processes for all IT services	Ensure compliance Increase transparency Reduce operating cost
	Correction of incidents as soon as possible	Increase management satisfaction Increase transparency Reduce operating cost
	Active IT demand management	Increase transparency Reduce operating cost
Supplier management	Better purchase prices and quality trough supplier consolidation	Ensure compliance Increase management satisfaction Reduce operating costs
	High quality of the suppliers	Improve capability provision
	Increase ratio of SLAs met by suppliers	Increase management satisfaction Increase transparency Reduce operating costs
	Limit the total number of suppliers (max. 20)	Ensure compliance Improve capability provision Increase homogeneity Increase transparency Reduce operating cost
	Sourcing through work packages	Ensure compliance Reduce operation cost Reduce security breaches
GRC	Ensure compliance to IT governance and control environment	Ensure compliance Increase transparency
	Adherence to architecture standards	Ensure compliance Increase transparency Reduce security breaches
	Reduction of application risks	Ensure compliance Reduce security breaches

Table 4.1.: BSC perspectives, goals, and mapping to BEAMS goals

Based on the mapping to the BEAMS goals, a valid starting point for the application of our method was given. Hence, in a designated workshop with one of the involved enterprise architects and one of the CIO office employees, the first step of our method (*characterize situation*) was performed. The result was the identification of 10 metrics recommended by the method base. In the second step of the method (*select & define metrics*), the previously identified metrics were examined. By executing the sub-step *select metrics*, the metrics - *SLAs met*, *incident duration*, *project performance index*, and *audit findings* were added to the result set of this method step for the further organization-specific configuration and instantiation step of the method. The remaining 6 metrics - *forecast quality*, *customer satisfaction index*, *employee satisfaction index*, *IT staff training*, *SLAs met by external suppliers*, and *Business applications compliment with IT architecture and technology standards* - were adapted during this workshop by executing the method sub-step *adapt metrics.* Thereby, either the information model and calculation rule of the recommended metrics, or the mapping to corresponding goals and concerns was changed. For instance, the metric *SLAs met by external suppliers* was defined based on the recommended metric *SLAs met* by the method base. In this case, the underlying information model (cf. Figure 4.1) was extended by the attribute *isInternal* to distinguish between internal suppliers (other IT departments of the organization) and external suppliers. Consequently, the calculation rule of the metric was adapted as well. The full transcript of the performed changes during this method sub step is available in Rassa's thesis [Ra13, p. 61-67]. For 11 of the predefined IT goals, no recommended metric was identified in the method base. Hence, the method sub step *define metrics* was executed 11 times. As described by Rassa, for five of these goals suitable metrics could be identified in the investigated BSC literature sources - *budget adherence* [SK04], *amount of innovation and R&D projects* [Tr10], *supplier's certification rate* [SK04], *number of suppliers for IT consulting and engineering* [SK04], and *approved application security concepts* [SK04]. These metrics were documented by the application of our MMFS structure and added to the existing result set. For the remaining 6 goals, in a separate 3 hours' workshop, 6 corresponding metrics were defined using the GQM approach of Basili et at. [BCR94], cf. Rassa [Ra13, p. 59-60]. These metrics have the following titles - *IT productivity according to 'top+' methodology*, *cost allocation value*, *amount of ITIL conform IT services*, *active demand management*, and *target supplier*, and *work packages share at purchasing volume.* Figure 4.2 illustrates the integrated information model (by merging the information models of all 21 metrics) for the developed metric system. Thereby, the model shows which elements from different EA layers and cross-cutting functions are taken into account by the defined metric system.

After documenting all 21 metrics by our MMFS structure, the results were presented to the CIO, the involved enterprise architects and CIO office employees. The experts agreed on the completeness of the proposed metric system and agreed with the execution of the third method step - *configure & implement metrics.* Thereby, in one further workshop with the corresponding stakeholders, the organization-specific configuration of the 21 metrics was performed, and all organization-specific description elements were documented in the MMFS instances, which are available in the Appendix section of Rassa's thesis [Ra13, p. 79-100]. For confidentially reasons, no names of the mapped organization-specific information sources, as well as no names the involved stakeholders (owner, responsible person, data contact) were documented. Nevertheless, the experts agreed with the publication of some of their organization-specific metric properties, which were correspondingly documented by us.

During the application of the method, two changes of out MMFS structure were performed by the involved experts. The MMFS structure was extended by the organization-specific description element *organizational goal.* The experts explained the need of this MMFS extension by the importance of documenting their precise organization-specific goals next to the related general EA management goals (cf. Figure 4.1). Additionally, the experts proposed

SLAs met by external suppliers

Management summary

Measure of SLA fulfillment by external suppliers.

Information model

SLA

name:string[1..1]
isInternal:boolean[1..1]
startOfPeriod:date[1..1]
endOfPeriod:date[1..1]

Goals

Ensure compliance
Foster innovation
Improve capability provision
Improve project execution
Increase disaster tolerance
Increase homogeneity
Increase management satisfaction
Increase transparency
Reduce operating cost
Reduce security breaches

Organization-specific instantiation

Mapping:

Name in model	Mapped name	Data contact	Data source
SLA			
name			
isInternal			
startOfPeriod			
endOfPeriod			

Properties:

Metric property	Property value	Observed values
Measurement frequency	Monthly	
Interpretation		
Owner	CIO	
Responsible	Supplier Manager	
Target value	100%	
Planned value(s)		
Tolerance value(s)		
Escalation rule		

Organizational goal

Increase management satisfaction, increase transparency, reduce operating cost and improve quality through increasing SLAs met through the external suppliers.

Calculation rule

Number of SLAs met by external suppliers divided by the total number of SLAs with external suppliers within a given time period.

Code

SMP-KPI-003

Source

Industry project

Layers

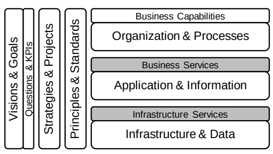


Figure 4.1.: Example of an adapted metric during the application of our method according to [Ra13]

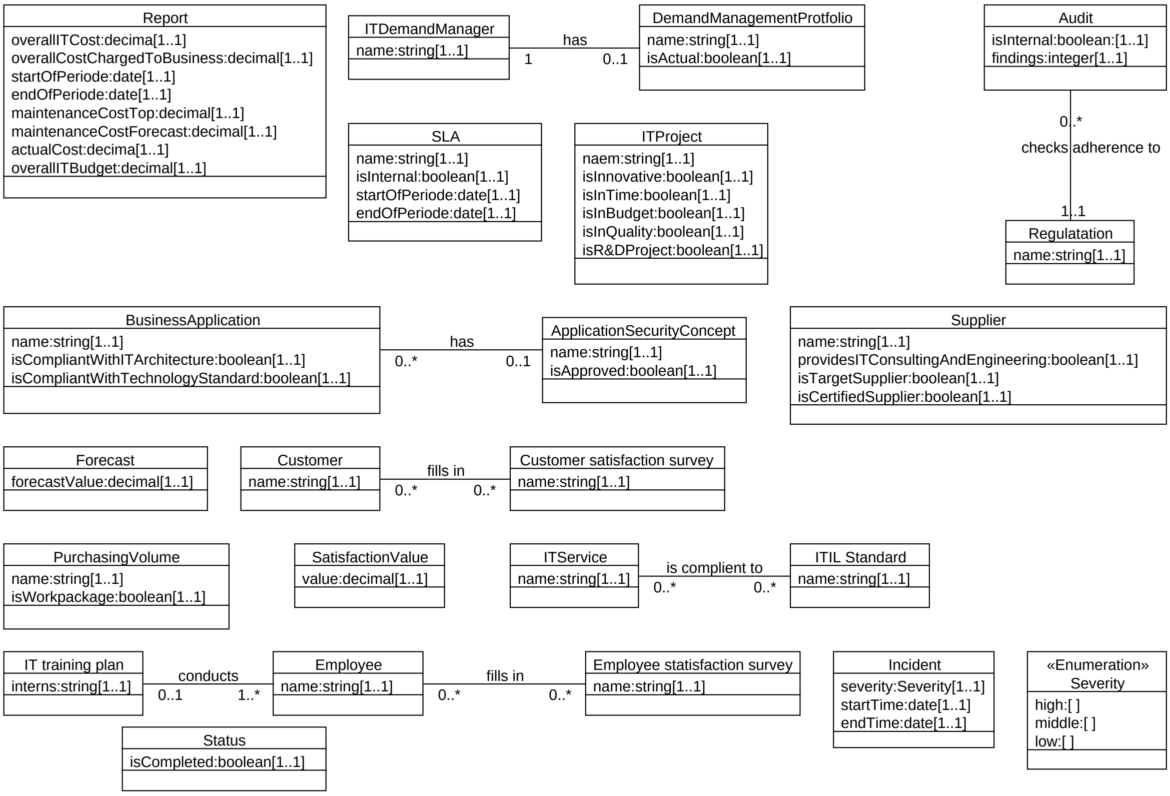


Figure 4.2.: Integrated information model of the metrics defined in the case study according to [Ra13]

the adjustment of the MMFS *code* element according to the corresponding BSC perspective. For example, metrics related to the *financial perspective* were documented with a code starting with the sequence *FP-KPI* and metrics related to the perspective *supplier management perspective* received a code starting with the sequence *SMP-KPI*.

After completing the case study, we asked the involved experts for an interview to collect feedback regarding our method. For this propose, one of the CIO office employees collected the feedback of all involved stakeholders and participated in a 45 minutes interview. In this interview, the expert confirmed that all stakeholders consider the method as helpful and suitable for the development of organization-specific metrics in the domain of EA management. Further, he reported that the expectations of all stakeholders were met by the method, and that the developed metric system is currently in implementation. Secondly, the expert reported, that the developed metric system allows goal-oriented and transparent monitoring and controlling. Moreover, he confirmed that the method base provides helpful knowledge and supports the timely retrieval of relevant metric best-practices. For the predefined IT goals, for which no relevant metrics were found in the method base, the expert emphasized the benefit provided by the method by performing a collaborative application of the GQM approach and collaborative documentation of metrics using our MMFS structure, which fosters a comprehensive understanding for all stakeholders.

Asked for additional improvements of the method design, the expert proposed the definition of a guideline for the proper usage of the MMFS structure. According to the expert, some of the involved stakeholders have experienced problems with the order, in which the MMFS structure is to be filled out. Based on his experience with the method, the expert proposed to start with the description of the general EA management goals and the *organizational goal* elements, followed by the documentation of the title, and management summary of a metric. Then, the calculation rule, concerns, code and source elements are to be defined. At the end, the information model should be defined followed by the documentation of the organization-specific mapping of the model and the documentation of the metric properties.

At the end of the interview, the expert reported, that his organization plans to use our method in the future, when new demands for new metrics arise. Further, the interviewee confirmed several challenges with the implementation of the developed metrics. Some of these metrics were already implemented by a corresponding *Excel* sheet, some other within the *SharePoint* system of the CIO office. A decision for one designated tool has not been made. According to the expert, different solution alternatives are in evaluation. Nevertheless, the expert confirmed the validity of our idea to perform metrics life-cycle management within one designated tool.

4.2. Critical Reflection

In Section 3.1, we presented four typical usage scenarios for metrics in the domain of EA management - *static EA assessment*, *EA change assessment supporting EA planning and controlling*, and *(enterprise) system behavior assessment*. The scenarios were extracted from the pool of related works for our domain (cf. Table 2.2). Hence, we are aware that one can doubt the completeness of these scenarios. Based on the reviewed literature sources, we can ensure that all described scenarios for the usage of quantitative models in these sources were incorporated in our solution and that we are not aware of further literature sources describing additional metric usage scenarios. Although, future research should investigate the completeness of the presented scenarios by collecting more empirical data, e.g. by expert interviews with both—practitioners as well as researchers in our domain.

Further, in Section 3.2 we presented a terminology base for the application of quantitative models in the EA management domain. Therefore, we examined the terminology used by the different author groups in the three investigated management domains (cf. Table 2.2, Table 2.3, and Table 2.4). We are aware that one can doubt the completeness of our results. Nevertheless, we showed that the existing terminology is too heterogeneous and hence, confusing in our understanding. Further, with respect to the presented metric usage scenarios, each of the terms used in the related literature can be linked either to the term *EA metric* or *performance indicator*. Thereby, our systemic perspective on an enterprise helps to distinguish between models concerned with the quantification of static aspects of a given EA and models quantifying specific behavioral aspects. Nevertheless, future research should conduct more empirical data regarding the completeness of our categorization as well as the proposed terms by conducting expert interviews with both—industry experts and scientists.

In Section 3.3, we presented a comprehensive collection of 26 risks and 39 recommended countermeasures for our domain. These risks and countermeasures were extracted from the related literature in all three investigated management fields, and their relevance, appropriateness, as well as completeness were confirmed by 19 industry experts. In addition, all risks were taken into account by our metric management method. Nevertheless, we are aware that more empirical data is required to justify the completeness and validity of our results. Therefore, future research should focus on the collection of additional empirical data for this purpose.

In Section 3.4, we presented a generic metric management fact sheet (MMFS) designed to support the definition and documentation process of metrics in our domain. Regarding the design of the proposed graphical representation of the structure, we accounted only for the positioning of the MMFS elements to fit on a single DIN A4 page in a way, allowing as much space for the documentation of the related content as possible. Nevertheless, we are aware, that different alternatives for the organization of the MMFS elements are possible, and hence, should be the subject of future research. For instance, the *Business Model Canvas* concept of Osterwalder [OP10] has gained a lot of popularity and acceptance in the business domain. Hence, the categories of the business model canvas could be adapted to the MMFS elements for the application in the EA management domain. This type of representation might be more intuitive for stakeholders from the upper business and IT management. Further, the representation of the MMFS could be adapted to the needs of the different involved stakeholders. For instance, the information model and its organization-specific mapping might be hidden in the representation for the upper management, but shown to metric owners, responsible persons, data contacts, and enterprise architects. Moreover, during the case study described in Section 4.1.2, we received the feedback that more guidance can be provided with respect to the order of filling out the different MMFS elements. Hence, future research should investigate benefits and disadvantages of different orders of filling out the MMFS structure. In this manner, a precise documentation of the process can be created to allow more efficient application of the MMFS structure. Further, during the evaluation of the MMFS, few experts reported that UML might not be the best suitable language to document the underlying model of a metric. As discussed previously, we doubt the suggestions that BPMN or i^* are more suitable for this purpose than UML. Nevertheless, future research should investigate the usage of other modeling languages, which are used in our domain, e.g. *ArchiMate 2.1* [Jo13b].

In Section 3.5 we presented an organized metric best-practice collection for our domain. Thereby, the catalog consists of 52 metrics documented by our MMFS structure and observed in German industry during the course of a research project. Additionally, the catalog provides navigation support for users to timely retrieve known best-practices according to their concerns (EA management goals and EA layers & cross-cutting functions). Nevertheless, we see following improvement potentials for this artifact as part of future research:

1. The metrics described in the catalog are observed only in German industry, mostly in the financial sector. In other industry branches and in other countries, additional metrics might be in use. Hence, future research should focus in the observation and documentation of additional metrics. Further, in our experience, metrics are considered as sensible piece of information in industry, hence, practitioners are not always prone to share their metrics with researchers. Thus, future research should focus primary on the documentation of general MMFS elements to ensure, that organization-specific and confidential data is not the subject of documentation.

2. Based on the low number of interactions with organizations, we can consider the documented metrics in the catalog only as pattern-candidates (cf. Ernst [Er10]) in the terminology of the pattern research community. Hence, the catalog users will benefit from the collection of more empirical data regarding the usage of concrete metrics. In this way,

metric patterns and metric anti-patterns for specific EA management problems might be observed and documented over time. Additionally, as described in Section 3.6, different versions of a given metric might be observed gradually and linked to specific organizational context elements allowing further improvement of the metric recommendations mechanism of the catalog.

3. As we experienced during several talks with scientists and industry experts, the question regarding the considered quality criteria of the catalog's metrics was frequently asked. Here, we want to clearly state, that the only quality criterion for all catalog's metrics is the observation of a metric in industry, hence we were not able to collected relevant data allowing the empirical analysis of additional metric quality criteria (cf. Jäger-Goy [JG02]), or even more - allowing sensitivity analysis of these metrics. Nevertheless, we explicitly described known risks and suggested countermeasures for the usage of metrics in our domain as a support for organizations interested in the application of our catalog. In our understanding, future research should focus on the collection of empirical metric usage data to enable the consideration of additional quality criteria for the catalog's metrics. Therefore, we encourage the practitioners using our catalog, in particular consultancies, to provide relevant (anonymized) data as the base for a corresponding research initiative.

Further, in Section 3.6, we presented a holistic life-cycle management method for metrics in the domain of EA management. The method was integrated within the BEAMS framework as an extension allowing the development and management of organization-specific metrics. The method uses our terminology, MMFS structure, catalog, and accounts for the risks presented in Section 3.3. However, as discussed previously, our method requires a software support integrated within an EA management tool. Hence, we consider the design and prototypical implementation of a corresponding prototype (cf. Chapter 5) as critical for the successful application of our method by user organizations. Further, the method was evaluated only in one case study. Consequently, future research should focus on the application of the method in further case studies to provide more empirical evidences for the correctness and completeness of the method design and to identify concrete improvement potentials.

In general, we see many benefits associated with the application of metrics in our domain. According to our experience, metrics help enterprise architects as well as their stakeholders to better understand specific aspects of their EA management functions, to identify concrete improvement potentials, and to quantify given evolution aspects of their EA. Nevertheless, as we already discussed, the application of metrics is related to several risks, which have to be taken into account. Therefore, we recommend to organizations interested in the usage of metrics, to not link too early metrics to the personal performance of the involved stakeholders, e.g. the responsible persons, as this decision can lead to several risks, e.g. the *meeting the numbers* phenomenon. In contrast, in a first step towards the definition of organization-specific metrics and the implementation of a corresponding metric management method, organizations should focus on the identification of a minimal set of EA metrics aligned with the needs of the relevant stakeholders to provide a quantitative assessment of the relevant EA elements. In a second step, organizations can use their metrics as additional support for the definition of target and planned EA states, as well as to assess performed EA changes. Further, organizations should consider the analysis of time-series to allow the usage of performance indicators, according to the needs of the involved stakeholder. Additionally, we encourage enterprises to build up a empirical metric usage data base, to allow the analysis of additional metric quality criteria, e.g. sensitivity analysis of metrics or correlation analysis. Furthermore, we want to emphasize the importance of understanding a given EA management metric management initiative as a holistic process. In this context, the process does not end with the implementation and regular measurement of metrics. The metrics have to be aligned with changing EA management goals and concerns in the organization. For this purpose, the used metric set has to be frequently revised, in particular when a related stakeholder changes. In addition, we recommend to account for a minimal number of metrics. In this context, we suggest organizations to try always to identify a metric deletion candidate, whenever a new metric is introduced in the given organization.

CHAPTER 5

Integrated Software Support

According to Chapter 1, organizations require specialized tools for their EA management initiatives. Over the years, a multitude of corresponding tools has emerged, which are used in industry. Thereby, the question of selecting the "right" tool for a given organization cannot be easily answered, since different implementation aspects need to be taken into account. To support the process of selecting an appropriate EA management tool for a given organization, our group published two EA management tool surveys - EAMTS 2005 [Se05] and EAMTS 2008 [Ma08]. Thereby, based on predefined EA management scenarios in collaboration with user organizations, different tools have been evaluated with respect to the degree of their support for the given scenarios. All of the investigated tools support the modeling of the underlying EAs, offer different techniques to gather and manage corresponding EA model (instance) data, and provide guidelines for the visualization of specific EA aspects. Nevertheless, as the results of the surveys confirm, the information structures and the collaboration mechanisms provided by the investigated tools are rather rigid, which yields to a problem in the EA management domain as described by Matthes et al. in [MN11]. Hence, our group proposed a wiki-based approach for EA management (called *Wiki4EAM*), allowing the incremental and collaborative enrichment of initially unstructured information sources (wiki pages) with structured information, e.g. attributes, types, relationships, and integrity rules.

5.1. Tricia

To prove the applicability of the previously presented wiki-based concept, our group created a prototypical software support called *Tricia*. Tricia is a Java-based enterprise 2.0 wiki system, which was initially developed at our chair and is now owned by the software company infoAsset AG [1]. According to Büchner et al. [BMN10], Tricia's main purpose is to support a collaborative

[1] `http://infoasset.de`

information management based on a flexible data model. In particular, the software system can be employed as a collaborative EA management tool, cf. Matthes et al. [MN11]. Tricia is based on the *Model-View-Controller* (MVC) pattern, cf. Starke [St14], i.e., the *view* of the web application (user interface) and its *model* (data) are decoupled and brought together by a *controller* component. This pattern ensures the separation of the user interface and the logic of the web application on the one hand, and facilitates the reuse of models and views on the other hand.

5.1.1. Models in Tricia

Tricia is designed to account for the usage of an arbitrary database management system (DBMS), where the applications data is stored. For this purpose, Tricia uses a flexible persistence layer to store the application data. Therefore, a so-called *model layer* abstracts Tricia's persistence layer from the underlying DBMS. Further, as depicted in Figure 5.1, Tricia supports a set of predefined *basic models*, where each of these basic models can be extended by inheritance. These basic models are defined as follows:

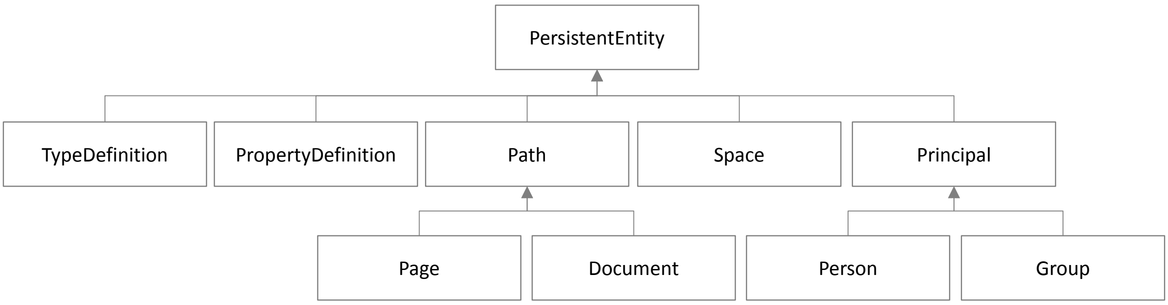


Figure 5.1.: UML class diagram of Tricia's basic model hierarchy according to [Re13]

TypeDefinition & PropertyDefinition *Type definitions* and *property definitions* allow the definition of information model's schema at runtime, hence we refer to these elements as schema objects in this work. Each type definition can be assigned to information objects (e.g., pages and documents) and contains an arbitrary number of property definitions. Further information regarding these elements is provided in Section 5.1.3.

Page & Document *Pages* are the main information objects in Tricia. Figure 5.2 gives an example for the description of this PhD thesis within the Tricia system used by our chair [2]. As illustrated in this figure, a page consists at least of a *name*, *tag(s)*, an *unstructured rich-text content*, as well as of a *type*, *attributes* and *relationships*. The attributes and relations of a page are either defined by the assigned type definition (by corresponding property definitions) or free attributes (i.e., arbitrary name-value-pairs attached to the page). Since pages have both—structured (type, attributes, and relations) and unstructured (rich-text content) data - we name them *Hybrid Pages* (cf. Subsection 5.1.3).

The concept of a *document* is very similar to concept of a page, i.e., a document consists

[2] `http://wwwmatthes.in.tum.de/pages/2ts6to3vritk/`

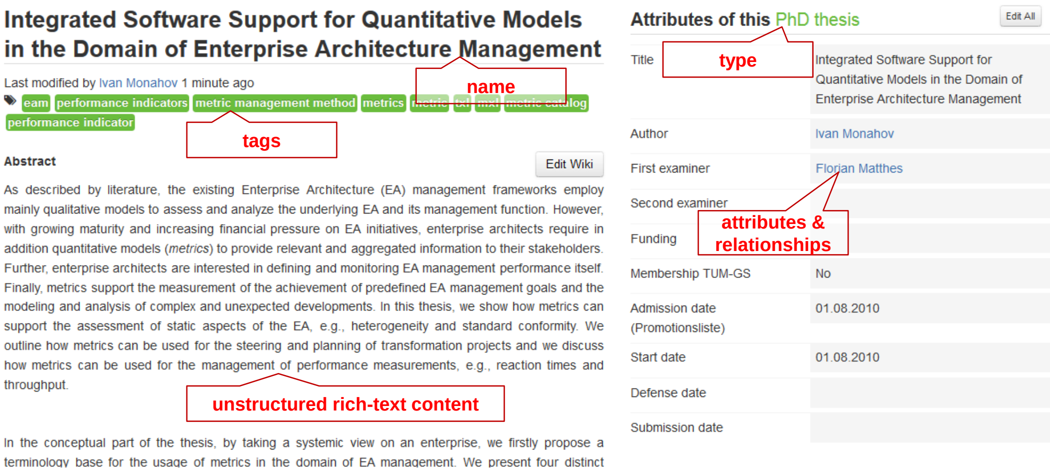


Figure 5.2.: A Tricia page and its basic parts according to [Re13]

of the same properties as described above. However, in contrast to pages, documents are directly related to (physical) files uploaded to Tricia.

Space *Spaces* are defined in Tricia as containers for pages, documents as well as for type definitions and property definitions, i.e., each information and schema object is part of exactly one space. Spaces are comparable to the concept of packages in Java and are defining an own name space, i.e., there may be two types with the same name in two different spaces, while all the types of one space require unique names. Furthermore, Tricia supports the import and export of spaces (including objects as well).

Person & Group Since Tricia is designed to support collaboration of multitude of users, an authentication and authorization mechanism is indispensable for the system. Hence, Tricia support firstly a common authentication process for its users (by user name and password). Secondly, the system manages two types of user access right for each Tricia object. A Tricia user has either *read-only* access to a certain Tricia object (the user is considered as a reader of the given object), or *write* access (the user is considered as a writer of a given object). Consequently, write access implies read access for a given pair of user and object.

5.1.2. Controllers and Views in Tricia

As a web application implementing the MVC pattern, Tricia's controller handles HTTP requests as described below, cf. Figure 5.3:

1. The web server is responsible for the authentication of users and the creation or recovery of sessions.
2. Based on the received URL, the web server forwards the request to a specific *handler* (controller).
3. The called handler checks, if the given user is allowed to perform the requested handlers action.

4. If the user is authorized to perform the action, the handler loads corresponding models and performs the associated business logic (e.g. update of corresponding objects)
5. Subsequently, the handler returns a corresponding view, which is presented to the requesting client. The view's layout and its design are thereby defined by a corresponding Tricia template associated to the view, whereas the presented content is instantiated by the view itself.

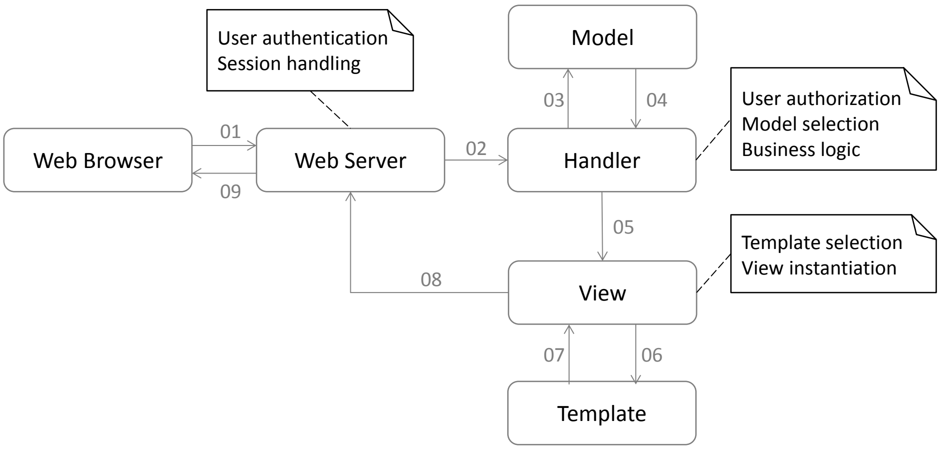


Figure 5.3.: HTTP request processing by Tricia according to [Re13]

5.1.3. Hybrid Wikis

The so-called *hybrid wikis* are one of the core concepts of Tricia. The term *hybrid* refers to an emergent enrichment of unstructured content (e.g. free text or documents, cf. the left part of Figure 5.2) with structure (types, attributes, and relationships, cf. the right part of Figure 5.2).

Attributes of type Employee

Attribute Name	Attribute Type	Multiplicity	Action
✎ Last name	Text	Exactly one value	Edit Delete
✎ Location	Reference Department	Exactly one value	Edit Delete
✎ Picture	Image	Exactly one value	Edit Delete
✎ Salary	Number	Exactly one value	Edit Delete

Figure 5.4.: Example of a type definition *Employee* consisting of four property definitions according to [Re13]

As previously described, two default models provided by Tricia are *type definition* and *property definition*. A type definition can consist of several property definitions, which in turn can define an arbitrary number of integrity rules:

Type If the property definition defines an attribute type, the attribute value of each of the type definition's instances has to be of this type, otherwise a warning is displayed in the instance. Tricia provides a basic set of attribute types, e.g. *Text*, *Number*, *Date*, *Boolean*, and *Reference* (a relationship to other instances, optional restricted to instances of a certain type). For instance, the property definition *Location* in Figure 5.4 is from type *Reference*, whereas the referred object has to be of type *Department*.

Multiplicity If the property definition defines a multiplicity, the attribute of each of the type definition's instances has to have the number of values as defined by the property definition, otherwise a warning is displayed in the instance. The multiplicities provided by Tricia are *Any number*, *At least one*, *Exactly one*, and *Maximal one*. For instance, all property definitions in Figure 5.4 are defined with the multiplicity *Exactly one*, hence each instance of type *Employee* has to provide exactly one value for each of its attributes.

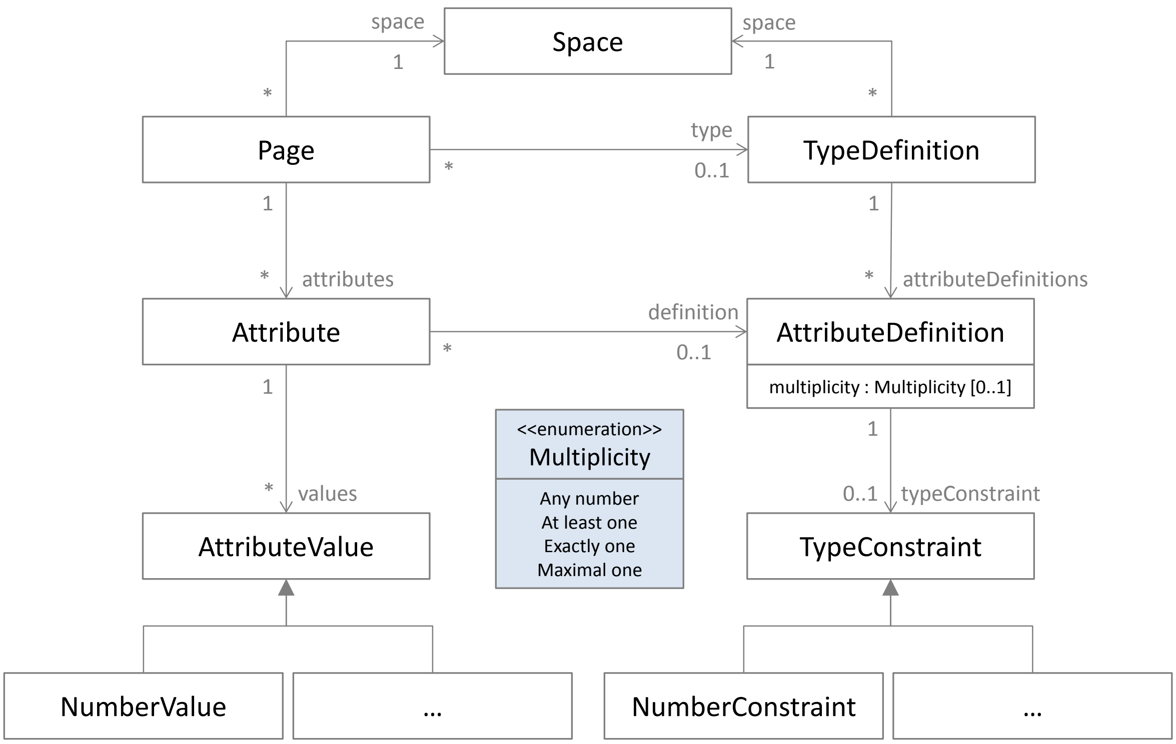


Figure 5.5.: UML class diagram of the abstract Hybrid wikis data model according Matthes et al. [MNS11].

The relationships between schema objects (type definitions and property definitions) and information objects (pages) are depicted in the hybrid wikis data model in Figure 5.5.

5.1.4. Outline of the Desired Integrated Software Solution

According to our research question 8 (cf. Section 1.1) and as discussed in Chapter 3, we consider the development of an integrated software support for our concepts as an essential part of our solution. Hence, in this chapter, we focus on the prototypical implementation of our conceptual solution. For this purpose, we decided to use Tricia as the foundation of the desired implementation and hence, we propose concrete extensions of the system by corresponding integrated capabilities to achieve our goal as described in the subsequent paragraphs.

Implementation of the MMFS concept According to Section 3.4, a metric instance is conceptually described by a corresponding MMFS instance. Hence, the MMFS structure can be implemented as a type definition in Tricia (we call this type **Metric Description**), cf. Figure 5.6. Thereby, all ten MMFS description elements can be implemented by

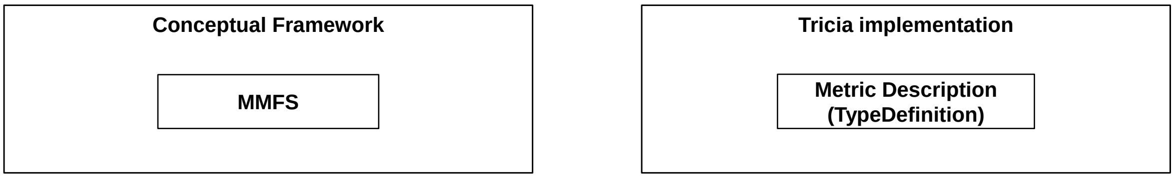


Figure 5.6.: Implementation of the MMFS structure

corresponding attribute definitions. Further, according to Neubert [Ne12], the underlying information model of a metric (by accounting for the organization-specific information model mapping of the given metric) can be implemented by the definition of a corresponding Hybrid model, cf. Figure 5.7. In addition, we extend the type *metric*

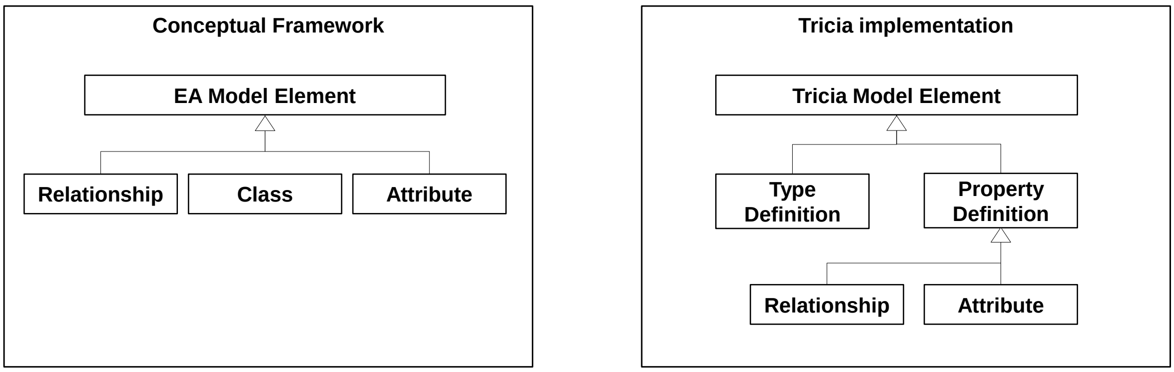


Figure 5.7.: Hybrid implementation of metrics information models

description by the hybrid attribute - **formula**. This element is used for the formal implementation of the underlying calculation rule of a metric by a model query, which can calculate the metric based on the given hybrid model and its instance data. This extension of the MMFS structure follows one of the recommendations for improvement of the MMFS structure gathered during the expert survey (cf. Section 3.4.5). Moreover, a view template, ensuring the proper graphical representation of the type *metric description* according to the graphical representation of the MMFS described in Section 3.4 is required. Thereby, the visualization of instances from type *metric description* requires a different visualization than the default representation of types in Tricia.

Implementation of the metric catalog The metric catalog (cf. Section 3.5) can be implemented as a Tricia space (workspace), cf. Figure 5.8.

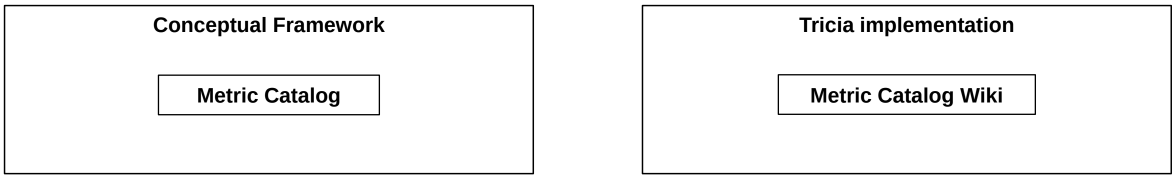


Figure 5.8.: Implementation of the metric catalog as a Tricia workspace

In this workspace, by using the predefined type *metric description*, all 52 metrics from the catalog can be implemented as Tricia pages of the type *metric description*. Additionally, both navigation aids from the catalog can be implemented within a corresponding navigation matrix (HTML table). Further, the catalog must provide a read access to all users. However, write access rights have to be granted only to members of our group to ensure managed evolution and consistency of the metric descriptions. Based on Tricia's *clone* functionality for workspaces and objects, interested organizations can clone either the complete catalog workspace or specific metric descriptions from the catalog workspace into their private Tricia workspaces. The management of access right for these private workspaces can be performed by the organizations themselves. In this way, the organization-specific adaption of the catalog's metrics can be ensured and their life-cycle can be managed in Tricia.

Definition of an EA management domain-specific model query language Based on the idea of applying model query languages to formally describe and calculate metrics in the modeling community, we require the extension of Tricia by a corresponding domain-specific model query language (we name this language *Model-based Expression Language* (MxL) in the remainder of the thesis). Further, in the subsequent subsection, we present concrete requirements that MxL has to fulfill. We name the language's prototypical *Tricia implementation* (TxL).

To ensure a sufficient software support for our metric management method, TxL has additionally to account for following functionalities:

Metric relationships management Based on the underlying MxL language design, TxL has to provide transparency of the relationships of metric instances at compile time. Thereby, for a given MxL query, both—MxL queries used by the given query and MxL queries using the given query - have to be accounted.

User-defined metric visualizations As described in Section 3.4, visualizations are a commonly accepted instrument to communicate metric results. In our understating, metric visualizations must be managed at metric instance level in the corresponding tool support, hence, TxL must provide mechanisms to support user-defined metric visualizations. To the given point in time, Tricia supports a variety of configurable software maps in terms of qualitative visualizations for Wiki4EAM. Nevertheless, a support for the visualization of quantitative models is not provided by Tricia.

User-friendly query definition support Since TxL is used as a web-based *integrated development environment* (IDE) for MxL queries, the prototype has to offer sufficient user support for the definition of queries in a web browser. Hence, TxL requires a *query editor* supporting *syntax highlighting*, *automatic completion* and *basic debugging* for MxL queries (by the provision of sound error messages for errors in MxL statements and a link to the corresponding MxL code line).

Accounting for these prototype requirements, a corresponding TxL implementation can provide the required support for a collaborative, adaptive, and adaptable life-cycle management of quantitative EA models in analogy to the existing support for qualitative models provided by the Hybrid wiki implementation.

5.2. MxL 1.0 Design and its TxL 1.0 Implementation

The first step towards the desired TxL implementation was done early in the year 2013, as published by us in [MRM13]. We decided to start the implementation with an initial MxL version (called **MxL 1.0**) and its initial integration within Tricia (**TxL 1.0**) as a *prove-of-concepts* implementation. For this purpose, we derive the following 8 requirements for MxL, which have to be fulfilled to ensure sufficient support of the desired software support:

Sufficient & minimal expressiveness MxL must provide sufficient and minimal expressiveness for the implementation of the catalog's metrics. Consequently, based on an analysis of the corresponding 52 calculation rules, we defined the 11 query operators - *where*, *take*, *select*, *selectMany*, *skip*, *concat*, *orderby*, *groupby*, *distinct*, *intersect*, and *except*) and the following 9 aggregation operators - *count*, *sum*, *min*, *max*, *average*, *first*, *firstOrNull*, *single*, and *aggregate*. More information regarding these operators in provided in Section 5.3.2. Further, we defined 8 the following basic types in MxL 1.0 - *string*, *number*, *boolean*, *date*, *sequence*, *map*, *function*, and *entity*. More information regarding these types are available in Section 5.3.1.

Higher-order functions Functional programming is characterized by the absence of side effects and in addition, it supports higher-order functions and recursion [VRH04, Sc01]. Higher-order functions are functions, which can take other functions as arguments. Hence, to support metrics aggregation, MxL must provide corresponding functional programming paradigms to support higher-order functions.

Lambda & implicit lambda expressions To provide more intuitive support for the syntax of expressions, MxL must support the definition of lambda expressions, i.e., the definition of anonymous function expressions. Further, by supporting implicit lambda expressions, MxL can empower more intuitive spelling of higher-order functions, e.g. expressions similar to the well-known *select-from-where* clauses of the *Structured Query Language* (SQL).

Access to model history According to Section 3.2, metrics have to provide reliable quantitative information regarding static aspects of the system's structure or performance aspects of its behavior over time. Hence, MxL has to provide corresponding capabilities to access the history of the underlying model.

Type safe MxL requires a type checker to resolve identifiers occurring in its expression and to check their types and semantics at compile time. In this way, MxL enables metric relationship management at run time in TxL and additionally supports the identification of affected MxL queries by changes in the underlying qualitative model.

Object-orientation Based on Tricia's design, its objects have types, attributes, and relations, an object-oriented language allows their representation by complex objects. Hence, accounting for corresponding object-oriented paradigms in MxL's design allows a convenient access to the information object's data.

Transitive closure Typical EA elements, e.g. *IT services* or *business processes*, are characterized by reflexive parent-child relationships. Hence, calculations of specific attributes of types with such reflexive relationships must be recursively performed. Therefore, MxL must provide a corresponding mechanism to support recursion.

Forward & backward navigation Tricia supports only directed relationships between types. Nevertheless, for specific metrics from the catalog (e.g. *Backuped key roles* [Ma12a, p. 20]), backward navigation is required for the calculation of these metrics. Otherwise, the underlying calculation rule cannot be implemented by only one MxL query (two separate queries are required for the implementation).

With respect to the MxL requirements presented above, we evaluated prominent model query languages with respect to their degree of support for our requirements. As described by us in [MRM13], we studied the two query languages - OCL [Ob12], which is widely accepted and used by different research modeling communities and Microsoft's general-purpose query language *LINQ* (cf. Box et al. [BH07]), which is widely accepted in research and industry.

Although each of these two languages fulfills the majority of our requirements, none of them fits ideally for our purpose. First, LINQ, as well as OCL, are general purpose programming languages, hence the number of offered types, operators and basic functions is oversized for the purpose of implementing the metrics of our catalog, as only a small set of these language constructs is sufficient for our needs.

Further, OCL is designed and optimized for its usage in the Eclipse environment. Nevertheless, as we target integration within the web-based EA management tool Tricia, the integration of OCL is not the best design decision in our understanding. Additionally, LINQ is designed and optimized for its usage in the Microsoft's environment. Since Tricia is Java-based, the integration of this language does not seem to be an optimal design alternative, too.

Further, in our understanding, an integrated domain-specific model query language in Tricia for the purpose of enabling a holistic metric life-cycle management has to be sequence-oriented. Hence, LINQ seems to be better suitable for our solution than OCL. In addition, JavaScript is widely used in the domain of web development. Nevertheless, in our understanding, the mixture of functional and object-oriented programming paradigms provided by LINQ and OCL are better suitable for our purpose, hence we decide to not use JavaScript.

Based on the discussed advantages and disadvantages of the related programming languages, we see Microsoft's LINQ language as the proffered solution for our problem, nevertheless, hence Tricia is a Java-based web application, we decide to not integrate LINQ, but to design and implement our own language, which however is strongly influenced by the concepts used in LINQ. By this design decision we can ensure that the benefits of LINQ's functional and object-oriented language features, as well as its sequence-orientation, are considered by our language. Further we can ensure a minimal number of types, functions and operators, as well as to account for the proper integration of our language in Tricia's code base.

For the initial implementation of MxL, we decided to account firstly only for the requirements *sufficient & minimal expressiveness*, *object-orientation*, *forward & backward navigation*, *higher-order functions*, *transitive closure*, and we decided to support lambda expressions (however, not yet implicit lambda) to timely obtain a working prototype to evaluate our overall implementation design. In addition, for the TxL 1.0 prototype, we decided to implement all previously described web-IDE functionalities - *query editor*, *syntax highlighting*, *automatic code completion* and *basic debugging*. The major part of the implementation was done by our student Thomas Reschenhofer in the course of his guided research, whom we supervised.

The TxL 1.0 prototype supports the following three use cases (cf. Monahov et al. [MRM13]):

Custom functions In order to reuse MxL 1.0 expressions, TxL 1.0 users can create so-called *custom functions*, e.g. for each of the catalog's metrics a custom function providing a formally definition of the metric's computation rule can be created (cf. Figure 5.9). To

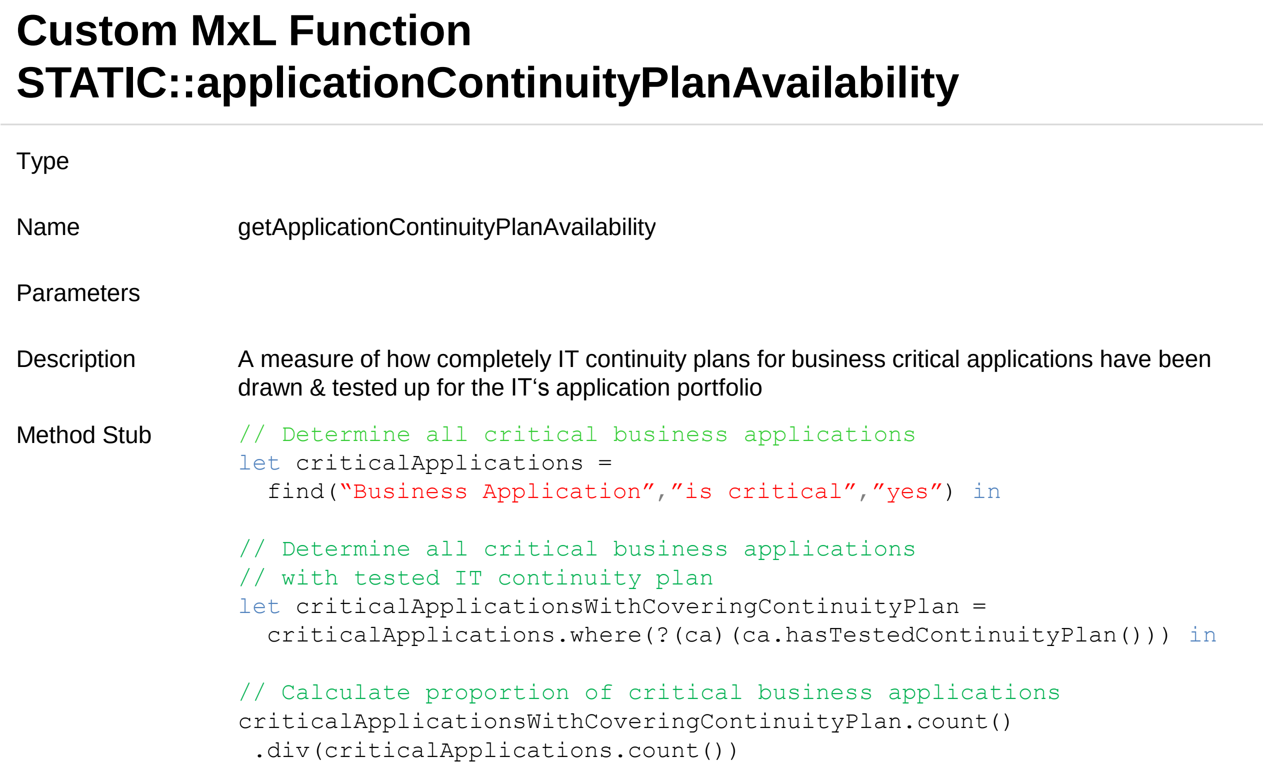


Figure 5.9.: Implementation of the calculation rule of the catalog's metric *Application continuity plan availability* [Ma12a, p. 19] as MxL 1.0 custom function

compute such a metric, the user has just to invoke the corresponding custom function.

Derived attributes While the values of common attributes are persisted by Tricia, the values of derived attributes are computed at runtime (by executing a corresponding MxL expression). Therefore, derived attributes make dependencies between information model elements explicit by the definition of corresponding MxL expressions. MxL expressions are consequently persisted by Tricia, but their values are evaluated at runtime.

Embedded expressions MxL 1.0 expressions can be embedded in the rich-text content of a Tricia page, which in addition enables a dynamic generation of HTML-based visualizations, e.g. based on a predefined conditions, the MxL 1.0 expression returns an HTML image displaying either a green, yellow, or red traffic light.

5.2.1. Evaluation of the TxL 1.0 Prototype

After finishing the implementation of TxL 1.0, we firstly evaluated the prototype regarding its support for the implementation of the 52 metrics of the catalog. Therefore, we successfully defined 52 distinct queries (one MxL query per calculation rule) within TxL 1.0 and we successfully performed the calculation of these queries based on test data defined by us.

Further, we employed TxL 1.0 in an EU project called *SmartNet Navigator* as described by Matheis [Ma13]. In this project, over 30 companies from German textile industry participate in a collaborative innovation management process. Although this project does not belong to the classical EA management domain, it contains all relevant challenges of a typical EA management project - high number of involved stakeholders, plenty of collaborative tasks and decisions, and gathering relevant information for the required decision making from different sources. In this context, a business-user-specific metric represented as visualization had been developed using TxL 1.0 to support the decision making process.

The so-called *SmartNet Navigator* is the automated generation of the visualization of a project's progress. In the SmartNet context, a project consists of tasks and meetings, which are assigned to several activity types (cf. Figure 5.10). Each activity type is in turn associated with a process phase as well as with a management activity type.

Based on project tasks and meetings, the SmartNet Navigator visualizes the progress of a project. Thereby, the *status* of the tasks and meetings is stepwise aggregated to an *activity type* status, then to a *module* status, a *process phase* status(a tuple of process phase and management activity), and finally to a *project status*. Figure 5.11 depicts an excerpt of the SmartNet Navigator of an exemplary project from the TxL based implementation. Thereby, the rows represent the management activity types (e.g. 'Planning'), whereas the columns represent the process phases (e.g. 'Creation of Ideas'). The cells represent the modules (e.g. 'Planning in phase 'I - Creation of Ideas"), and the items in the cells represent the activity types (e.g. 'Identification of problems, needs, and opportunities'). Further, the status of an element is indicated by its color as defined in Table 5.1. For more detailed description of the evaluation of the SmartNet Navigator we refer to Hauder et al. [Ha13].

Based on the model illustrated in Figure 5.10 and an appropriate test data set, we defined custom TxL functions for the stepwise visualization and aggregation of a project's status. For example, the computation of a process phase's status is depicted in Figure 5.12, which in turn

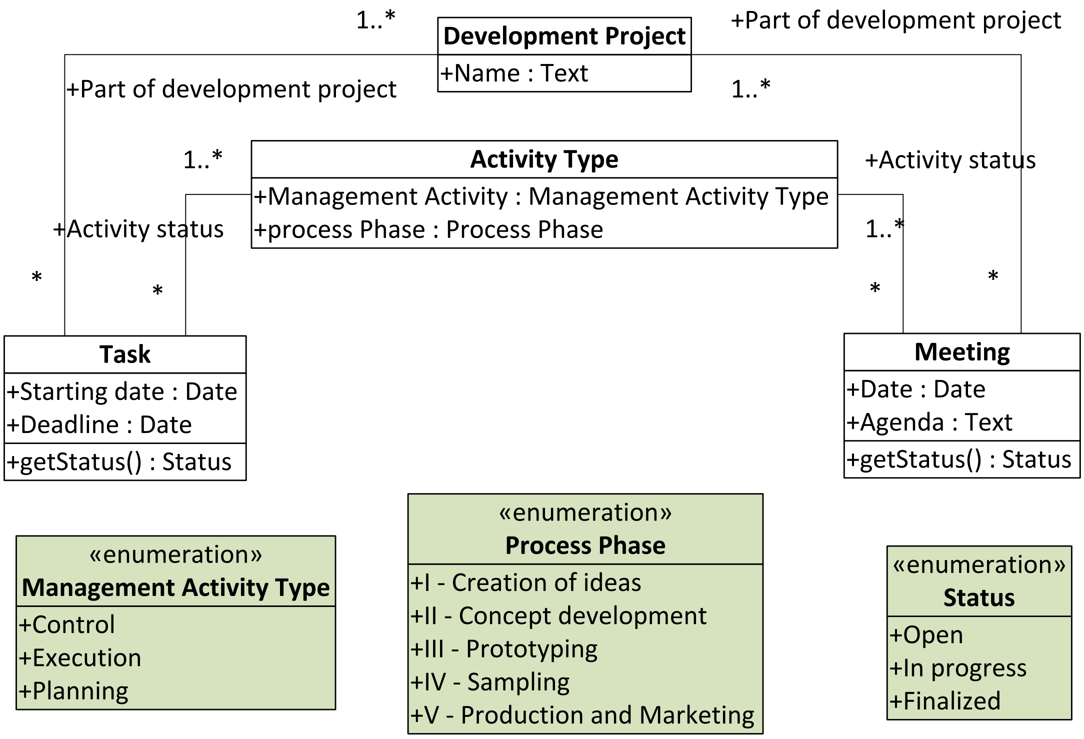


Figure 5.10.: An excerpt of the SmartNet information model according to [MRM13]

Color	Status	Aggregation
Grey	*Open*	If the status of each sub-element of an element is *Open*, the elements status is *Open* as well
Green	*Finalized*	If the status of each sub-element of an element is *Finalized*, the elements status is *Finalized* as well
Orange	*In progress*	If the status of an element is neither *Open* nor *Finalized*, it is *In progress*

Table 5.1.: Color encoding and status aggregations of a project, a process phase, a module, and an activity type

is used for the definition of the status table's header (e.g. by specifying the background color of the column). By executing the function *statusTable* on an element of type *development project*, the function generates the HTML markup defining the SmartNet Navigator (see Figure 5.11), which can be embedded in any Tricia page. The execution of the status aggregation and the visualization functions are depicted in Figure 5.13.

As this evaluation shows, the prototype was successfully used to define the underlying qualitative SmartNet data model as well as to define a corresponding quantitative model by using TxL custom functions at runtime. Further, TxL 1.0 supported the definition of complex visualizations and computations at runtime. In addition, this experiment was that successful and

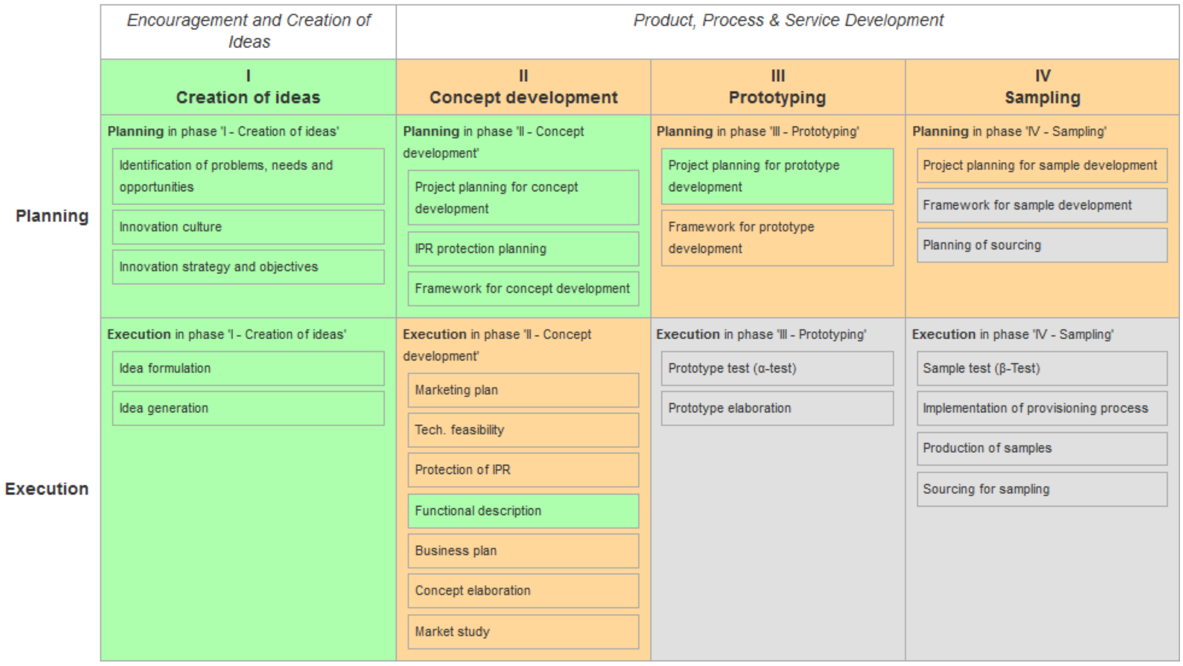


Figure 5.11.: An excerpt of the SmartNet Navigator of an exemplary project according to [MRM13]

Custom TxL Function Development Project::statusOfProcessPhase

Type	Development Project
Name	statusOfProcessPhase
Parameters	phase
Description	Returns the status of the given process phase in the current project by aggregating the status of all activity types of this phase

Method Stub

```
/* Determination of activity types for the given process phase,
   followed by the computation of each activity's status in the
   current project*/
let allStatus = find("Activity Type", "Process phase", phase)
  .select(?(at) (this.statusOfActivityType(at))) in

/* Definition of a helper function to combine two stati
let combine = ?(s1,s2) (
    s1.equals("finalized").and(s2.equals("finalized"))
      ? "finalized"
      : (s1.equals("open").and(s2.equals("open"))
          ? "open"
          : "in-progress")) in

/* Aggregation of the status list */
allStatus.aggregate(combine, allStatus.first())
```

Figure 5.12.: TxL 1.0 implementation of a process phase's status according to [MRM13]

useful for the involved business users, that the company infoAsset AG decided to adapt our prototype in their industrial version of Tricia.

Further, we applied our TxL 1.0 prototype in a parallel research project (called *CALM*[3]) at

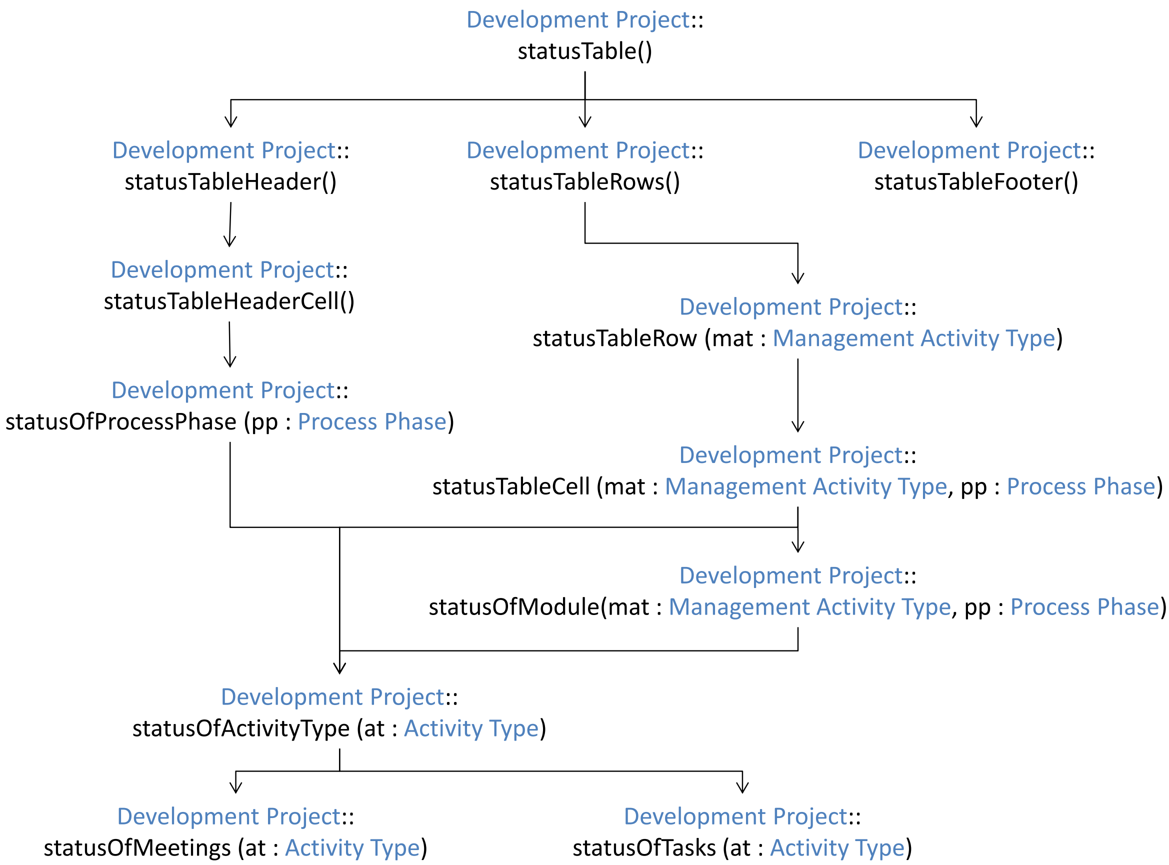


Figure 5.13.: Execution of the queries and visualization functions of the SmartNet Navigator according to [MRM13]

our chair, concerned with the measurement and management of complexity in *application landscapes* (AL). In this project [3], 6 organizations from the financial sector, 2 consultancies, 1 healthcare organization, and 1 car manufacturing organization participated in the identification of suitable metrics for the given problem. During a series of workshops, 15 metrics for the quantification of complexity of ALs by research literature, e.g. , topology-based metrics, cf. Schütz et al. [SWK13] and heterogeneity-focused metrics, cf. Lagerström et al. [La13] were investigated applied to the domain of AL complexity. In the course of this project, we firstly asked the participants for concrete metrics they are using for their EA management initiatives in order to extend the metric catalog and to evaluate the prototype with respect to the implementation of these metrics. Based on the collected feedback, we documented 21 new metrics in the Tricia version of our catalog [4]. We were able to implement all of these 21 metrics using the TxL 1.0 prototype. Secondly, the participants in the CALM^3 project decided to use TxL 1.0 as the metric management software environment for the set of 15 complexity metrics, which were defined in corresponding workshops. Thereby, all participants agreed to provide anonymized EA qualitative data as a basis for the calculation of the metrics. In addition, the

[3] http://wwwmatthes.in.tum.de/pages/1cfwykj33dfxb/
[4] http://131.159.30.135/pages/8wk3nt1cdms4/

participants agreed with benchmarking their metric results based on this anonymized data. Consequently, we started the TxL documentation and implementation of these 15 complexity metrics. However, during this implementation, we realized that TxL 1.0 does not provide required aggregation functions for the implementation of these metrics. More precisely, MxL 1.0 does not implement *exponentiation* and *logarithm* functions, which are required for the implementation of specific complexity metrics. Hence, an improved version of MxL must provide corresponding aggregation functions to empower the implementation of these metrics. In addition, several of these complexity metrics require recursion for their calculation.

Furthermore, during a demonstration session of TxL 1.0 in a CALM^3 workshop, one of the participating organizations - a mid-sized and internationally operating German public bank, agreed to apply an improved TxL prototype (TxL 2.0) for the life-cycle management of their organization-specific EA management metrics, cf. Chapter 6.

5.2.2. Shortcomings of TxL 1.0

According to the feedback collected during the application of TxL 1.0, as well as according to our implementation requirements, we identified the following 7 shortcomings of our prototype, which need to be addressed by an improved version MxL 2.0 of our DSL and by an improved prototype - TxL 2.0:

Insufficient expressiveness To fulfill the requirement *sufficient & minimal expressiveness*, MxL 2.0 has to support both—exponentiation and logarithm functions.

No compile-time analysis of expressions Since MxL 1.0 does not have an integrated type checker, the language is not type safe and hence, the requirement *type safe* is not fulfilled. Therefore, changes in the underlying qualitative EA model at runtime can lead to integrity violations of associated metrics. For example, according to Figure 5.9, the type *Business Application* might be renamed or the attribute *is critical* might be deleted by a user organization during the course of the organization-specific instantiation of the exemplary metric. TxL 1.0 is not able to recognize the expression's invalidity until a re-evaluation, which leads to a corresponding runtime exception, is performed. Further, a type checker offered by MxL can be used by TxL to enable metric relationship management at runtime. Consequently, MxL 2.0 must be extended by a type checker.

Tight coupling between MxL 1.0 and TxL 1.0 According to discussions with experts in the CALM^3 workshop, organizations might be interested in the adoption of MxL in their EA management tools. Nevertheless, in TxL 1.0, MxL and Tricia are tightly coupled, hence, the integration of MxL in other EA management tools was considered as complicated by the experts. Hence, MxL 2.0 must ensure loose coupling to EA management tools for the purpose of its integration.

Missing type-based template engine in Tricia To facilitate the representation of metric descriptions according to our visual representation of the MMFS structure, the layout and the design of metric descriptions (instances of the type metric description) have to be customized according to our template (cf. Section 3.4).

Missing implicit lambda support Although MxL 1.0 supports lambda expressions, according to the feedback of some of the CALM^3 participants, the support of implicit lambda expressions will empower the spelling of more intuitive expressions. Hence, MxL 2.0 must fulfill the requirement *Lambda & implicit lambda expressions*.

Missing access to model history Although only one of the CALM[3] participants provided historicization data for the EA of his financial organization, all participants confirmed the importance of considering model history by the defined complexity metrics. Further, the experts confirmed the added value for their management activities provided by the visualization of metrics accounting for historicization as means of trends recognition. Hence, MxL 2.0 must provide corresponding support for accessing model historicization data and hence, it has to fulfill the requirement *access to model history*.

Missing inheritance support Although MxL 1.0 supports object-oriented concepts, it does not support one of the fundamental concepts - *inheritance*. Hence, in MxL 1.0, reuse is supported only by the concept of delegation, cf. van Roy et al. [VRH04]. Nevertheless, as we experienced during the application of TxL 1.0, supporting inheritance will ensure a more convenient usage of MxL for the underlying object-oriented concepts offered by Tricia (types, attributes, and relationships).

User-defined metrics visualization According to the feedback from the CALM[3] workshop, user-defined metrics visualizations of metrics are considered as an important instrument for the communication of metric results to managers and colleagues. Hence, TxL 2.0 has to support user-defined metric visualizations (based on MxL expressions).

Following the main idea of design science research and based on the positive feedback collected during the application of our preliminary prototype, we decided to redesign MxL 1.0 and to improve our prototype according to the identified shortcomings described above. Therefore, in the second half of 2013, we firstly presented an improved MxL 2.0 version and a corresponding TxL 2.0 implementation. The main part of the implementation was done by Thomas Reschenhofer in the course of his Master's Thesis (cf. [Re13]), which we supervised at this time. Hence, a comprehensive description of MxL's 2.0 design and its prototypical implementation TxL 2.0 are available in [Re13]. Nevertheless, in the subsequent section we describe the most interesting implementation aspects and provide examples to support the understanding of the implementation.

5.3. MxL 2.0 Design and its TxL 2.0 Implementation

In this section we describe selected aspects from the design and implementation of MxL 2.0 and its implementation - TxL 2.0. Therefore, we firstly describe the types supported by MxL 2.0 in Section 5.3.1. Then, we present all operators supported by our prototype in Section 5.3.2, followed by a description of the interpretation process of MxL 2.0 expressions in Section 5.3.3.

5.3.1. Types in MxL 2.0

While MxL 1.0 does not support inheritance (cf. Section 5.2.2), MxL 2.0 supports this fundamental concept of the object-orientation paradigm for the purpose of reusing functionality. However, the basic type hierarchy of MxL 2.0 types is rather simple, since each type is derived from the type *Object* (except the type *Object* itself). Table 5.2 describes all MxL 2.0 types. Thereby, compared to MxL 1.0, three additional basic types are supported by MxL 2.0 – *Object*, *Type*, and *Space*.

Name	**Description**
Object	Each element of MxL's underlying information model is of type *Object*.
String	Each character sequence encapsulated in quotation marks is a value of type *String*, e.g. "Hello, World!".
Number	Represents both integers and decimals, e.g., *1.23*.
Boolean	*true* and *false* as well as the language-specific key words "yes" and "no".
Date	A date consisting of day, month, and year. A date can be constructed by the date-function and the date's string representation. The current date can be determined by the global identifier *Today*. The components of a date are accessible via *day* (e.g. *Today.day*), *month* (e.g. *Today.month*), and *year* (e.g. *Today.year*).
Map	A fixed collection of *key-value-pairs*. The notation is similar to the *JavaScript Object Notation* (JSON), e.g. *{title: "hello world"}*.
Entity	An entity is a complex object, e.g. an object with attributes and relations to other entities.
Sequence	An ordered multi-set of values, using the notation *[element1, element2]*. An ordered multi-set thereby describes a collection, whose order matters and additionally allows duplicates. The type *Sequence* can be parameterized to determine the type of the sequence elements, e.g. the type *Sequence<Number>* defines a sequence of numbers. The elements of a sequence are accessible via *[ ]* and the element's index (the index is zero-based).
Function	Hence MxL 2.0 supports higher-order functions, the language supports the basic type *Function*. This type can be parameterized to determine the function's signature (parameter types and return type), e.g. the type *Function<Number, Number, Boolean>* defines a function with two input parameters of type *Number* and returns an object of type *Boolean*. Furthermore, parameter types can be defined as *optional* by using a question mark (the function can be invoked without optional parameters), e.g. *Function<Number, Number?, Boolean>* can be invoked for either one or two parameters.
Type	A meta-type for the general representation of types in MxL, e.g. the types *Number* and *String* inherit from the type *Type*.
Space	Representation for a Tricia workspace (package consisting of types, static functions, and instances. This concept corresponds to the concept of *packages* in Java.

Table 5.2.: Basic types in MxL 2.0

Further, this set of basic types can be extended by a given MxL 2.0 implementation. For instance, the improved version of our prototype - TxL 2.0 - extends the MxL 2.0 basic types set by the concepts *Page*, *Document*, *Principal*, *Person*, and *Group*, according to Tricia's underlying architecture as described in Section 5.1.1. Thereby, each of these additional three types is derived from type *Entity*. Figure 5.14 shows an overview of the resulting type hierarchy.

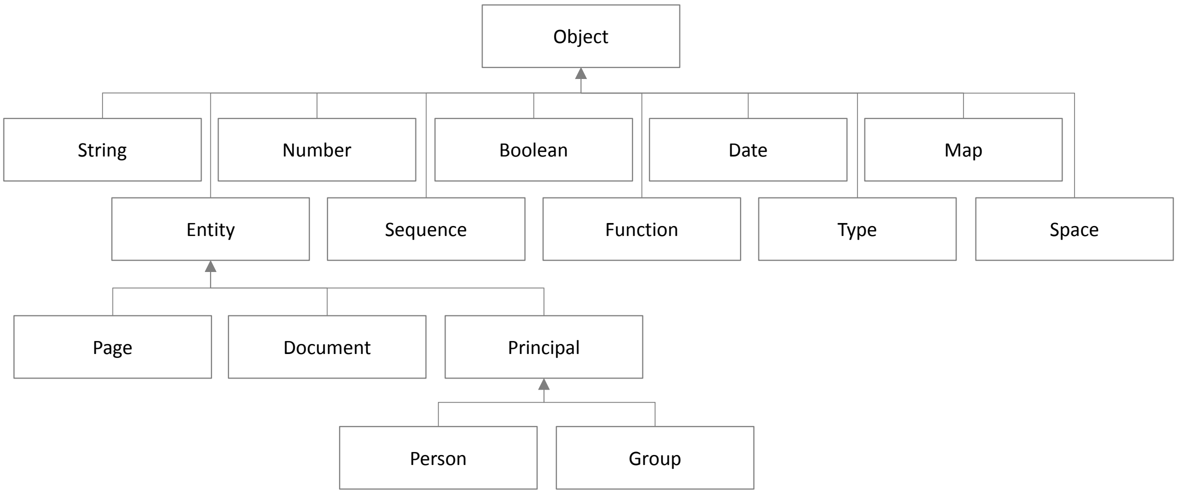


Figure 5.14.: Type hierarchy of MxL 2.0 for its integration in Tricia according to [Re13]

5.3.2. Operators in MxL 2.0

To provide sufficient support for the metrics implementation of the (extended) metric catalog, MxL 2.0 defines the following operators:

Arithmetic operators are supported to implement required arithmetic functions as depicted in Table 5.3. Further, the + operator can be used also for string concatenation, and the - operator supports the calculation of dates differences, e.g. *Today - date("01.01.2014")*.

Function	Operator	Example
Addition	+	1.0 + 2.0
Subtraction	-	3.0 - 4.8
Multiplication	*	3.14 * 2.72
Division	/	1.0 / 3.14
Exponentiation	ˆ	2 ˆ 8

Table 5.3.: Arithmetic operators in MxL 2.0

Comparison and logical operators Table 5.4 provides an overview of the corresponding operators supported by MxL 2.0:

Conditional operators In MxL 2.0, the syntax changed from the ternary operator (*<condition> ? <ifbrach> : <elsebranch>*) to an *if-than-else* statement:

```
if <condition>
then <ifbranch>
else <elsebranch>
```

Function	**Operator**	**Example**
Equality	=	1 = 1
Inequality	<>	2 <> 3
Greater than	>	3 > 1
Greater than or equal	>=	1 >= 2
Less than	<	2 < 4
Less than or equal	<=	3 <= 3
Logical inversion	not	*not true*
Conjunction	and	*true and false*
Disjunction	or	*true or false*

Table 5.4.: Comparison and logical operators in MxL 2.0

Sequence functions MxL is mainly used to define and execute queries against the underlying information model. Hence, MxL 2.0 needs to apply specific filters, projections, or aggregation on sequences of corresponding type instances. Hence, MxL 2.0 provides the common query functions (cf. Table 5.5), quantifier functions (cf. Table 5.6), set functions (cf. Table 5.7), element functions (cf. Table 5.8), partitioning functions (cf. Table 5.9), and aggregation functions (cf. Table 5.10) to ensure sufficient (and minimal) expressiveness power required to implement the metrics from the metric catalog. Please note, that all functions in these Tables are applied on sequences of type *Sequence*<*T*>. Further *T* and *V* are arbitrary MxL 2.0 types.

Name	**Parameters & Return type**	**Description**
select	map : Function<T, V> *returns* : Sequence<V>	Applies the map-function to each source sequence element and returns a sequence containing the results of each single application.
selectMany	map : Function<T, Sequence<V> *returns* : Sequence<V>	Similar to the select-function, however, in selectMany, the map-function returns a sequence for each element. The concatenation of all sequences forms the result of the selectMany-function.
where	pred : Function<T, Boolean> *returns* : Sequence<T>	Filters the source sequence by the given predicate, i.e., all elements fulfilling the predicate remain in the sequence.
groupby	keySel : Function<T, Object> f : Function<Sequence<T>, Object>? *returns* : Map	Groups the elements of the source list by the keySel-Function and applies the (optional) f-function on the elements of each single group.

Name	**Parameters & Return type**	**Description**
orderby	keySel : Function<T, Object>? descending : Boolean? *returns* : Sequence<T>	Sorts the source sequence by the (optional) keySel-function, whereas a natural order will be applied. The (optional) descending parameter determines, if the elements should be ordered ascending or descending.

Table 5.5.: Common query functions in MxL 2.0.

Name	**Parameters & Return type**	Description
any	pred : Function<T, Boolean> *returns* : Boolean	Returns *true*, if at least one element of the source sequence fulfills the given predicate, otherwise *false*.
all	pred : Function<T, Boolean> *returns* : Boolean	Returns *true*, if each element of the source sequence fulfills the given predicate, otherwise *false*.
none	pred : Function<T, Boolean> *returns* : Boolean	Returns *true*, if no element of the source sequence fulfills the given predicate, otherwise *false*.
contains	element : T *returns* : Boolean	Returns *true*, if the given element is contained in the source sequence, otherwise *false*.
isEmpty	*returns* : Boolean	Returns *true*, if the source sequence has no elements, otherwise *false*.
isNotEmpty	*returns* : Boolean	Returns *true*, if the source sequence has at least one element, otherwise *false*.

Table 5.6.: Quantifier functions in MxL 2.0

Name	**Parameters & Return type**	**Description**
distinct	*returns* : Sequence<T>	Removes all duplicates of the source sequence.
except	other : Sequence<T> *returns* : Sequence<T>	Returns a sequence with all elements contained in the source sequence, but not in the other one.
intersect	other : Sequence<T> *returns* : Sequence<T>	Returns a sequence with all elements contained in the source sequence and in the other one.

Name	Parameters & Return type	Description
concat	other : Sequence<T> *returns* : Sequence<T>	Concatenates the source sequence with the other one, i.e., the resulting sequence contains all elements of the source sequence, followed by all elements of the other one.

Table 5.7.: Set functions in MxL 2.0

Name	Parameters & Return type	Description
first	pred : Function<T, Boolean>? *returns* : T	Returns the first element of the source sequence (or the first element satisfying the predicate). If there is not such an element, this function throws an exception.
last	pred : Function<T, Boolean>? *returns* : T	Returns the last element of the source sequence (or the last element satisfying the predicate). If there is not such an element, this function throws an exception.
single	pred : Function<T, Boolean>? *returns* : T	Returns the only element of the source sequence (or the only element satisfying the predicate). If there is not such element, or if there is more than one element, the function throws an exception.

Table 5.8.: Element functions in MxL 2.0.

Name	Parameters & Return type	Description
rest	*returns* : Sequence<T>	Returns the source sequence without the first element
take	n : Number *returns* : Sequence<T>	Returns a sequence with the first *n* elements of the source sequence
takeWhile	pred : Function<T, Boolean> *returns* : Sequence<T>	Returns all elements of the source sequence until an element does not satisfy the predicate
skip	n : Number *returns* : Sequence<T>	Returns a sequence without the first *n* elements of the source sequence
skipWhile	pred : Function<T, Boolean> *returns* : Sequence<T>	Skips all elements of the source sequence as long as these elements satisfy the predicate, and returns the rest

Table 5.9.: Partitioning functions in MxL 2.0.

Name	Parameters & Return type	Description
count	pred : Function<T, Boolean>? *returns* : Number	Counts all elements of the source sequence (or counts the elements satisfying the predicate).
ratio	pred : Function<T, Boolean> *returns* : Number	Returns a number between 0 and 1 representing the ratio of elements fulfilling the given predicate.
sum	map : Function<T, Number>? *returns* : Number	Sums up all numbers of the source sequence. The optional map-function may select a numerical member of each element.
average	map : Function<T, Number>? *returns* : Number	Computes the average of all numbers of the source sequence. The optional map-function may select a numerical member of each element.
max	map : Function<T, Object>? *returns* : T	Determines the maximal element of the source sequence. The optional map-function may select a criterion used for the selection of the maximum.
min	map : Function<T, Object>? *returns* : T	Determines the minimal element of the source sequence. The optional map-function may select a criterion used for the selection of the minimum.
aggregate	func : Function<V, T, V> seed : V *returns* : V	This is a fold-operator aggregating the current sequence to a single value by the given *func*-function. The func-function is invoked for the result of its previous invocation and each of the source sequence's elements. The seed value is used for the first iteration of the func-function . The result of its last invocation is the result of the *aggregate*-function.

Table 5.10.: Aggregation functions in MxL 2.0.

Further information regarding additional MxL language constructs, e.g. *higher-order functions*, *lambda expressions*, and *name-value bindings*, is available in Reschenhofer's thesis [Re13]. Figure 5.15 summarized the hierarchy of the presented MxL 2.0 elements.

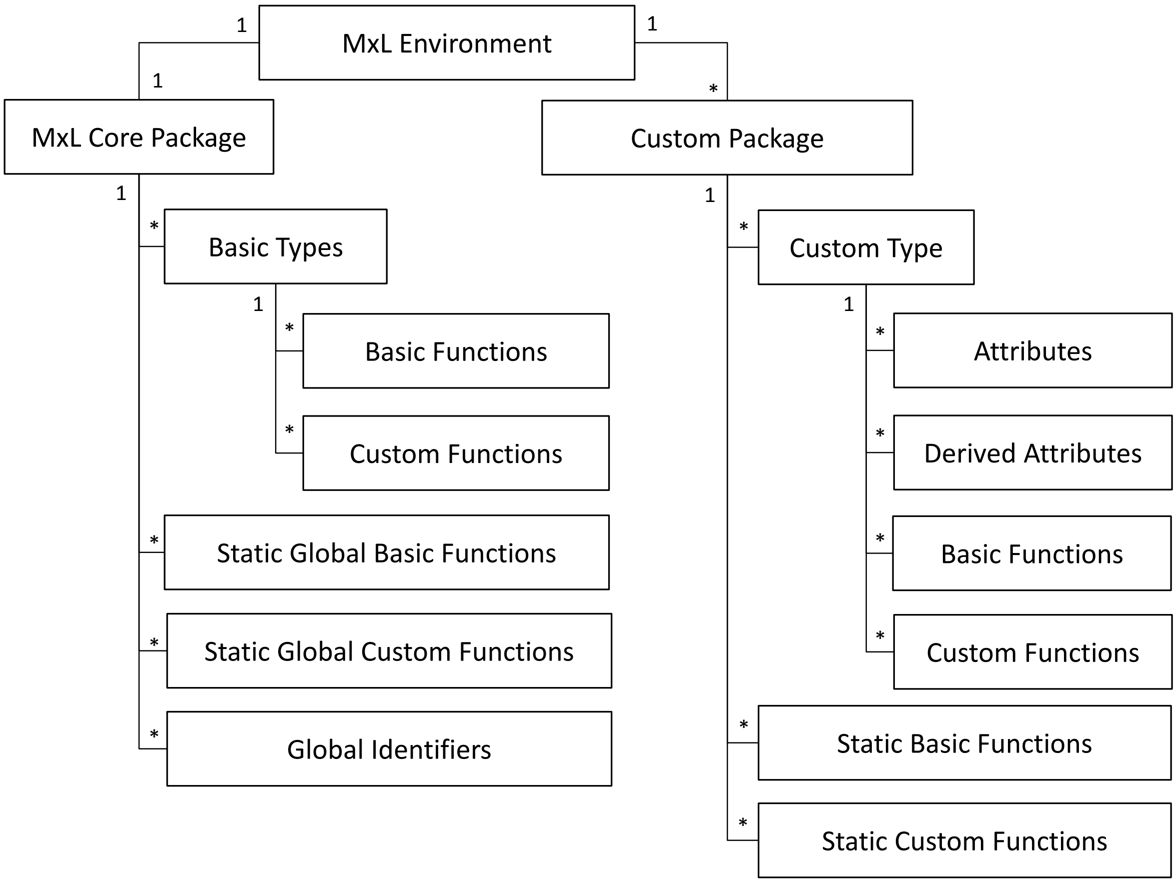


Figure 5.15.: The hierarchy of MxL's 2.0 elements according to [Re13]

5.3.3. MxL 2.0 Interpreter

Figure 5.16 illustrates the process of interpreting and evaluation MxL 2.0 expressions. In contrast to MxL 1.0, the elements *MxL type checker*, *MxL connector*, and *Schema* are introduced by MxL 2.0 according to its design requirements.

Scanner & Parser The scanner and the parser of MxL 2.0 did not changed significantly compared to MxL 1.0. The input of the scanner is a MxL expression represented as a stream of characters. The declaratively specified MxL 2.0 lexical grammar defines how characters are bundled to proper tokens, e.g. multiple digits to one number. The MxL 2.0 scanner was created based on the free and Java-based lexical analyzer generator JFlex (for more information regarding JFlex, we refer to the manual of Klein [Kl10]). Further, the tokens generated by the scanner are used as input for the parser of MxL 2.0, which creates an *abstract syntax tree* (AST). In analogy to its scanner, the MxL 2.0 parser was generated based on a declarative specification of MxL 2.0's syntax in *Extended Backus-Naur Form* (EBNF) [Sc93] and using the open source *LALR* [Ah06] parser generator called *Beaver* (for more information we refer to Demenchuk [De06]). Additional information regarding the processing of MxL's AST is available in [Re13, p. 40-42]

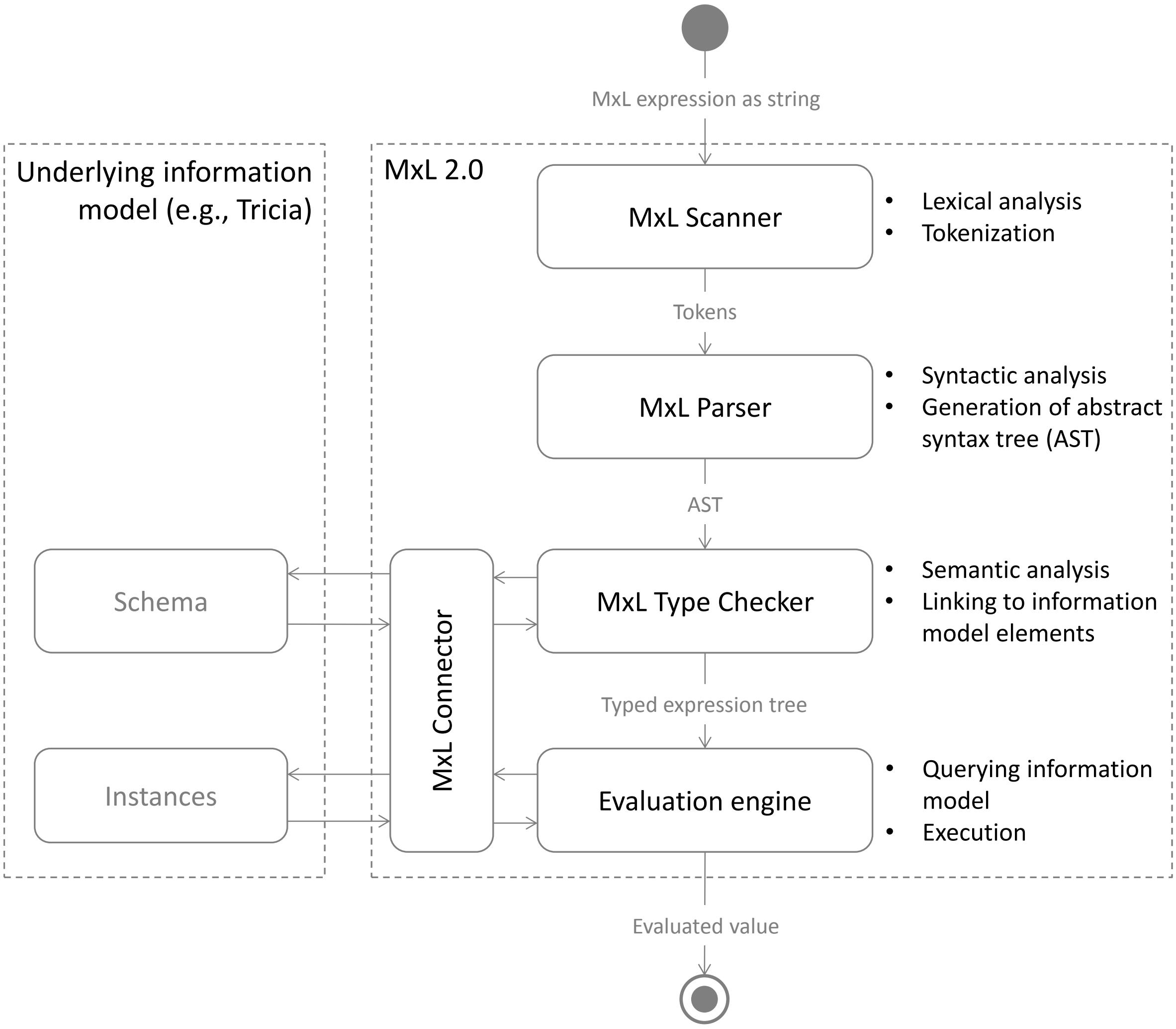


Figure 5.16.: Overview of MxL's 2.0 interpreting and evaluating process according to [Re13]

Type checker In contrast to MxL 1.0, MxL 2.0 implements a type checker. This component takes the AST from the parser and performs a type check on the root expression object and triggers a cascade type check on the entire AST. Thereby, the operation triggered by the type checker depends mostly on the specific type of the give expression. Trivial expressions, e.g. *NumberExpressions* do not require type checking, hence their types are fixed. Nevertheless, operator expressions, e.g. *arithmetic operation*, mostly depend on the type of their operands. For instance, the + operator can either perform a numerical addition, if the two operands are numbers, or string concatenation, if the operands are from type string. The UML activity diagram illustrated in Figure 5.17 explains the type checking process of the *FunctionParameterTypeChecker* implemented in MxL 2.0. In addition, this type checker implements the required feature *implicit lambda*. Thereby, if the standard type check fails, the checker ties to resolve the problem by the application of implicit lambda interpretation. If this operation is successfully, a type check of the interpreted lambda expression is performed and returned. Otherwise, the component

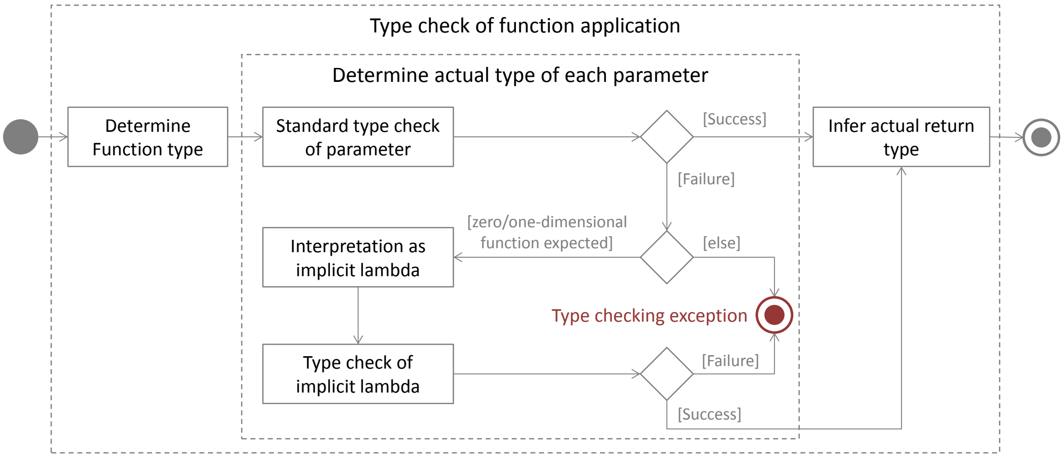


Figure 5.17.: Overview of the type checking process (*FunctionParameterTypeChecker* according to [Re13])

returns a type checking expression. A more detailed description of the type checker's implementation, as well as the hierarchy of supported exception types is presented by Reschenhofer [Re13, p. 42-48].

MxL connector To support more loose coupling of MxL within a given EA management tool for integration, we designed a corresponding MxL connector component as depicted in Figure 5.18. While MxL's 2.0 scanner and parser are rather autonomous component,

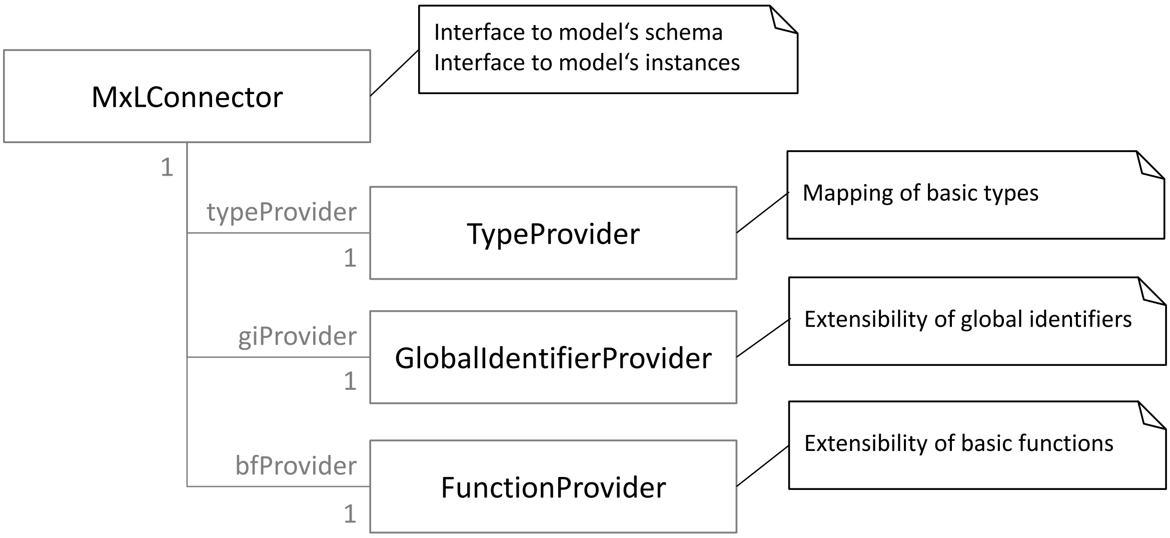


Figure 5.18.: The MxL connector and subcomponents according to [Re13]

both—MxL's type checker and evaluation engine require interactions with the underlying EA model. Hence the MxL 2.0 information model depends on the concrete MxL 2.0 implementation, the interpreter consists of a component called *MxL connector*. This component abstracts the interaction between MxL and the information model of its implementation. In this way, the connector defines an interface between MxL 2.0 and its implementing system. The MxL connector is a new component in MxL 2.0 and facilitates the implementation of MxL in arbitrary EA management tools. The connector defines following three methods:

Access to information model The connector provides a multitude of methods for gathering an information model's schema data (e.g. to get an attribute by its name and owner type) as well as its instances (e.g. get attribute value by its name and owner object).

Mapping of basic types As described previously, MxL 2.0 provides a set of 11 basic types. Consequently, these types have to be supported by each implementation of MxL 2.0. To accomplish this task, the MxL connector maps the types of the implementing EA management tool to MxL 2.0 basic types. This mapping is defined and managed by the connector's sub-component *TypeProvider*, cf. Figure 5.18.

Extensions of global identifiers While MxL 2.0 provides already a (minimalistic) set of global identifiers (e.g. *Today*), the MxL connector allows the provision of implementation-specific global identifiers. This extensibility is defined by the connector's sub-component *GlobalIdentifierProvider*.

Extensions of basic functions In analogy to the global identifiers, MxL 2.0 provides a set of different functions, e.g. basic and sequence functions as described above. Nevertheless, it might be necessary to extend this set by EA management tool specific functions. Therefore, the *FunctionProvider* subcomponent allows the definition and management of corresponding extensions of MxL functions.

Evaluation engine Hence MxL 2.0 provides a type checker, its evaluation engine has changed significantly compared to MxL 1.0. Hence, the evaluation is initiated at the AST's root and cascaded through the whole tree. Based on the usage of the type checker, the evaluation of expressions in MxL 2.0 is rather simple. The majority of the expressions can either be directly expressed by corresponding Java expressions (e.g. arithmetic and logical operations) or can be evaluated by calling the MxL executor (cf. Reschenhofer [Re13]).

Further, to identify and prevent infinite loops as well as to localize *MxLEvaluationExceptions*, the evaluation engine manages a stack trace. Each time the evaluation engine executes a function (or a derived attribute), it pushes the function's or derived attribute's identifier onto a call stack. If the execution of the function or derived attribute is completed, the evaluation engine takes off the call stack's upper element. The stack trace can be understood as a snapshot of the call stack. However, if an *MxLEvaluationException* occurs during the evaluation of a given expression, the stack trace represents a path to the source of the exception. Moreover, since the call stack also contains context parameters, the evaluation engine is able to check if a function (or derived attribute), is already evaluating with the current parameters, what indicates an infinite loop. In this case, the evaluation engine stops the evaluation and throws a corresponding exception.

5.4. Selected Implementation Aspects of TxL 2.0

In this subsection, we present selected implementation aspects of TxL 2.0. Therefore, we firstly describe how derived attributes and custom functions are implemented in TxL 2.0 (cf. Section 5.4.1). Then, we describe how TxL 2.0 empowers the relationships management between expressions by compile-time analysis in Section 5.4.2. Further, we present a so-called *at* operator enabling the access of TxL 2.0 queries to the model history of Tricia elements in Section 5.4.3. Furthermore, we describe a Tricia extension by the *Highcharts* library to enable user-defined metric visualizations at runtime in Section 5.4.4. In Section 5.4.5, we present selected capabilities of our prototype, designed and implemented to provide web-based query editing support to the users. Additionally, we describe how the MMFS structure is implemented in Tricia in Section 5.4.6.

5.4.1. Derived Attributes and Custom Functions

Based on the design of MxL 2.0, custom functions play an essential role for the desired software support. As previously discussed, the computation rules of metrics are implemented as MxL 2.0 custom functions. These functions are implemented as derived attributes in Tricia, which is a new construct introduced during the integration of MxL 2.0 and Tricia. Thereby, derived attributes and custom functions are added to Tricia's system base. The corresponding expressions are persisted by the system, nevertheless, these expressions are interpreted and evaluated first at runtime. Figure 5.19 illustrates the integration of these concepts in Tricia. In this example, the settings view of the type Employee lists all derived attributes, e.g. *Costs*, in a corresponding table under the hybrid attributes of the type. By clicking the link of a derived attribute, the corresponding type definition is shown, cf. Figure 5.20. Derived attributes are represented in Tricia similarly to hybrid attributes, cf. Figure 5.19.

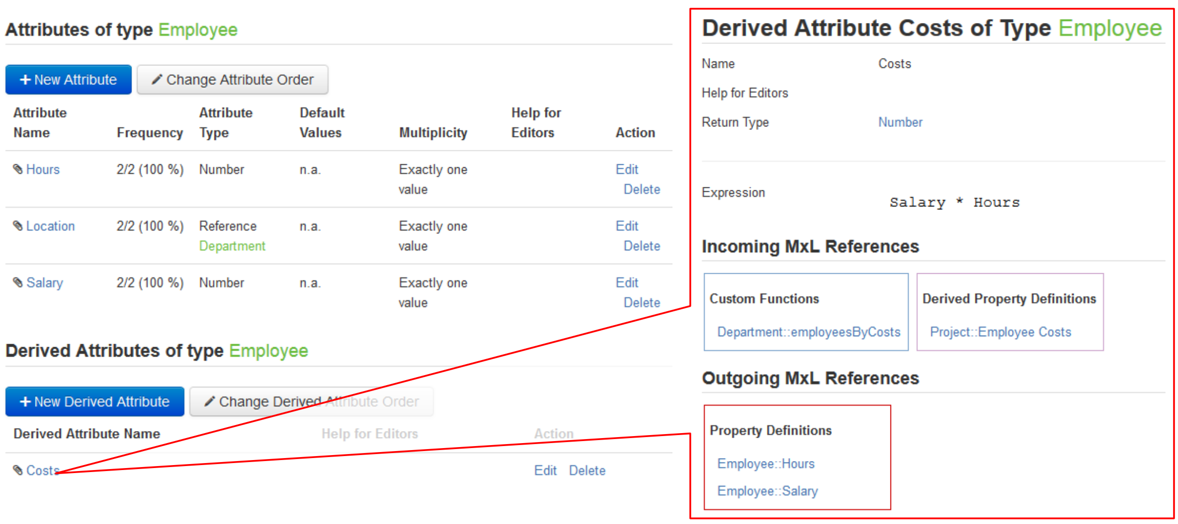


Figure 5.19.: Representation of derived attributes in Tricia according to [Re13]

Nevertheless, based on their different behavior, derived attributes are shown in a separate table. By clicking the link associated with a derived attribute of a given type, a corresponding representation of the derived attribute definition is called (cf. the right part of Figure 5.19). In this view, all properties of the derived attribute definition are shown, and can be edited.

In contrast to derived attributes, the custom functions of a given type (as well as all basic functions) are listed on a separate view of this type called *Functions*. The corresponding view provides a button for the creation of new custom functions as well as hyperlinks for editing and deleting. By selecting a custom function of a given Type (cf. Figure 5.20), its name, description, parameters, and method stub are shown as well as an inferred return type and all references to other MxL elements as described in the subsequent subsection.

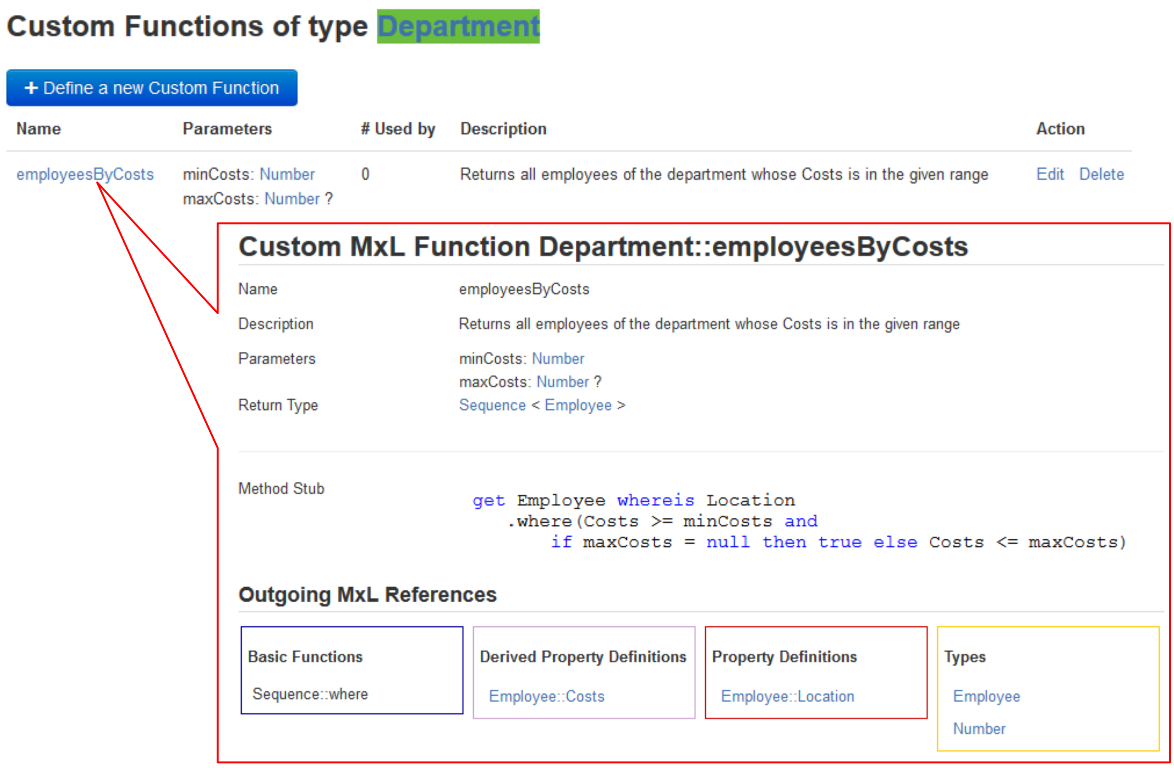


Figure 5.20.: Tricia's view for the management of derived attributes and custom functions according to [Re13]

5.4.2. Compile-time Analysis of Expressions

One of the major improvements of MxL 2.0 was the capability of a compile-time analysis of a MxL expressions and hence, the provision of a sufficient support for metrics relationship management at run time. Thereby, the analysis of a MxL expression relies on the determination of all MxL elements (cf. Figure 5.15), to which the identifiers of the expression refer to. By the design and the implementation of the MxL type checker, MxL expressions are correspondingly analyzable. Consequently, as described by Reschenhofer [Re13], the TxL implementation accounts for this idea as well. Figure 5.20 provides an example of this functionality. Thereby, by opening the management view of the custom expression *employeesByCosts*, Tricia shows all relationships to MxL elements in corresponding tables in the bottom of the view. In this example, the given custom function uses the basic MxL function *where*, depends on the types *Employee*, and *Number*, the derived attribute *cost* of the type employee and the attribute *Location* of the type employee. Figure 5.19 illustrates in addition an example for *outgoing MxL references*, where the derived attribute *Costs* is used by the MxL custom function *employeesByCost* and in addition, by the derived attribute *Employee Costs* from the type *project*.

By the provided implementation, TxL 2.0 supports an *adaptable* and *adaptive* metric relationship management as published by us in [RMM14] (please note, that the paper is currently in status *submitted for publication*). The term *adaptable* refers to Tricia's capability to allow users to define, change and delete MxL expressions (metrics) at runtime according to their needs. In addition, the system gives an immediate feedback to its users regarding the impact of their actions on related MxL expressions (by taking incoming references from other custom functions into account). The term *adaptive* refers to Tricia's capability to help users to understand an manage changes in related MxL expressions based on performed changes in the underlying model. For instance (cf. Figure 5.19), if the attribute *hours* from the type *Employee* is renamed by a user, all incoming MxL references are determined by the system and updated correspondingly (e.g. in the MxL expression of the incoming reference from the derived attribute *costs*). Further, if the attribute *hours* is to be deleted, the system requests a decision from the user, hence integrity constraints will be violated. Thereby, the user is requested to select one of the following two options:

Cascade deletion In this case, all MxL custom functions and derived attributes depending on the element *hours* will be deleted as well to prevent inconsistencies in the quantitative model.

Do nothing In this case, the user decides to cause inconsistencies in the corresponding MxL elements. Hence, the next time when one of the affected MxL elements is evaluated, an exception will be thrown by the system.

5.4.3. Access to Model History

To provide a prototypical implementation for history access of Tricia model elements, we supervised Manoj Mahabaleshwar in a corresponding guided research project [Ma14]. Thereby, we defined and proposed an extension of TxL 2.0 by the so-called *at* operator (notated as @). Thereby, this operator can be applied on an arbitrary TxL expression to perform the evaluation of the given expression based on Tricia's model elements values valid for a given point in time as an argument for the @ operator.

To provide an example for the usage of this operator, we define the following TxL expression as depicted in Figure 5.21. Thereby, the expression computes the number of employees, who

```
find (Employee).where (salary > 3000).count()
```

Figure 5.21.: Example of a TxL 2.0 query

have a salary higher than 3.000 per month. If this query is evaluated in TxL, only the current state of the model elements history is evaluated. Now, we assume that a managers requires the development of this expression over time based on the historicization of the underlying Tricia model elements and their instances. More precisely, the manager requires the exact expression values for the third and fourth quarters of 2013 (we refer to these two dates as to *Q3 2013* and *Q4 2013*), the first quarter of 2014 (we refer to this date as to *Q1 2014*) and the current value on the expression (*Today*). By using the @ operator, the manager can extend the expression as depicted in Figure 5.22.

```
find (Employee).where (salary > 3000)
  .count()@["2013-09-31", "2013-12-31", "2014-03-31", Today]
```

Figure 5.22.: Example of a TxL 2.0 query using the @ operator

Figure 5.23 provides an example of *change sets* for the history of a type *Employee* and its instances. The evaluation of the expression using the @ operator is performed as follows:

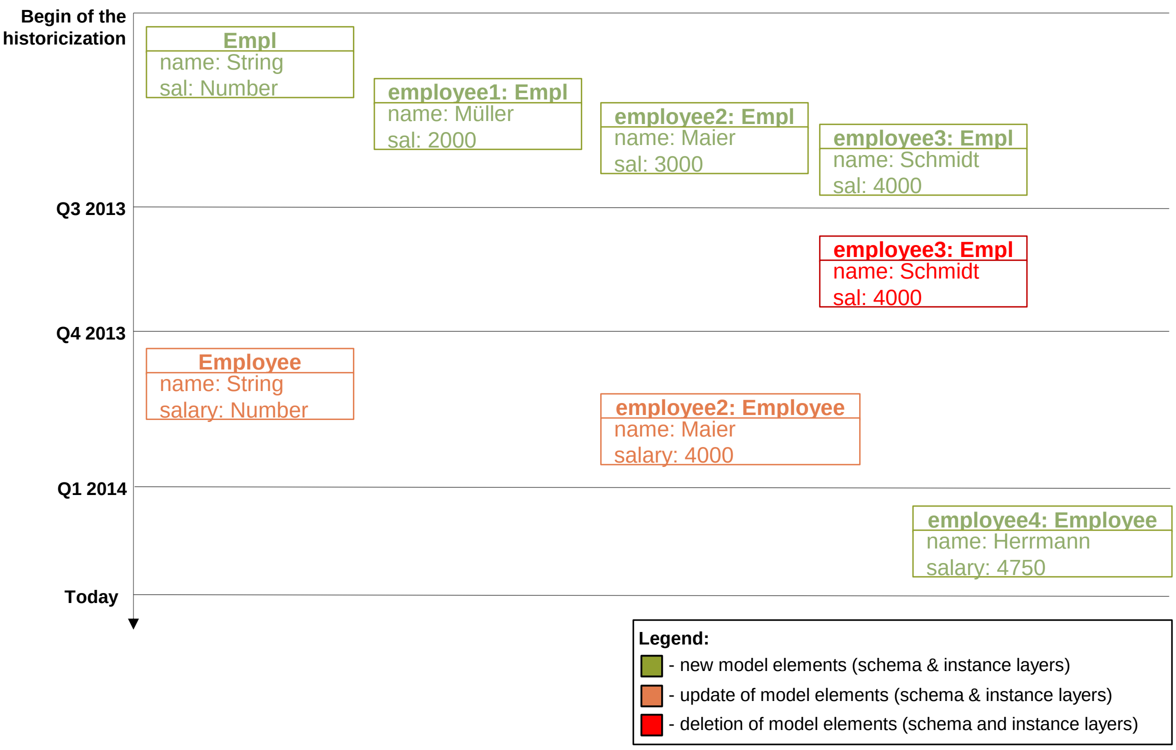


Figure 5.23.: Example for historicization change sets of a Tricia model according to [Ma14]

1. According to this example, at time point *Q3 2013*, the type *Employee* was initially named *Empl* and contained the two attributes *name* and *sal*. Further, at the given time, three instances of this type existed in Tricia – *employee1*, *employee2*, and *employee3*, which were created sequentially, according to Figure 5.23. The evaluation of the expression for this point in time returns consequently the result *1* (only the instance *employee3* fulfills the given criterion).

2. At time point *Q4 2013*, TxL detects that the instance *employee3* was deleted in the mean-

while, whereas the other two instances and the type *Empl* remain unchanged. Hence, the evaluation of the query for the given point in time returns the value *0* (none of the two instances fulfills the given criterion).

3. At time point *Q1 2014*, TxL detects that the type *Empl* was renamed to *Employee*, and the attribute *sal* was renamed to *salary*. Further, TxL detects changes in the value of the attribute *salary* for the instance *employee2* (the salary of the corresponding employee has changed from 3.000 to 4.000). Hence, the evaluation of the expression for the given point in time returns the value *1* (hence now employee2 fulfills the given criterion).

4. Applied at the current state of the model (application of the @ is not required, or can be parameterized with *@Today*) , TxL detects that a new instance of type *Employee* was defined in the meanwhile - *employee4*. Hence, the evaluation of the query returns the value *2* (employee2 and employee4 fulfill the given criterion).

5. The result of the evaluation of the exemplary query is the sequence *[1, 0, 1, 2]*.

As this example shows, the prototypical TxL implementation of the @ operator supports the access of Tricia's model history by TxL 2.0 queries. In addition, this operator enables the visualization of metric history in corresponding metric visualizations, cf. Section 5.4.4. Nevertheless, the proposed prototypical implementation faces several limitations, e.g. missing history import functionality in Tricia, as described in Section 7. Additional information regarding the @ operator can be found in the work of Mahabaleshwar [Ma14].

5.4.4. User-defined Metric Visualizations

To enable user-defined metric visualizations, we supervised Michael Schätzlein in a corresponding Master's thesis project [Sc14]. Based on a literature review concerned with the identification of typical metric visualizations in the fields of *IT controlling*, *enterprise controlling*, *financial controlling*, *business intelligence* and *information visualizations*, we proposed a set of 5 visualization types for the metrics from our catalog as described below:

Line charts A *line chart* (cf. Figure 5.24) is a visualization which displays a series of data as points connected by straight lines.It shows how the observed metric value changes over time. As a consequence, the x-axis has to consist of consecutive date values. Related to line charts are also the so-called *area charts*, where the area below the line is colored. For the visualization of metrics, line and area charts can be utilized to show trends and patterns in the value of a metric over time. Figure 5.24 shows an exemplary visualization for the metric *Application continuity plan availability* from our catalog as a line chart, whereas Figure 5.25 shows a visualization of the same metric as an area chart. Further, line charts can be used to visualize several metrics at the same time. Therefore, all metrics require the same result type, otherwise additional y-axes are required. According to Tufte [Tu01], this might lead to confusion and suggests patterns where none exist.

Column charts This type of visualization shows a series of data as rectangular bars with lengths proportional to the represented value. The bars can be plotted vertically (*column charts*) or horizontally (*bar charts*). Column charts can be utilized to either display a snapshot of a metric with multiple values as result type (e.g. the metric *Incident duration* from our catalog [Ma12a], cf. Figure 5.26), whereas bar charts (cf. Figure 5.27) are

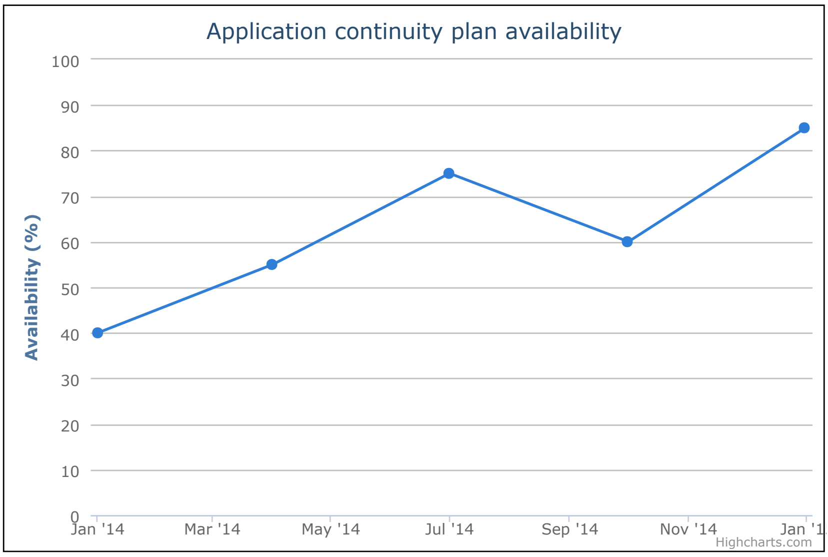


Figure 5.24.: Example of a line chart visualization according to [Sc14]

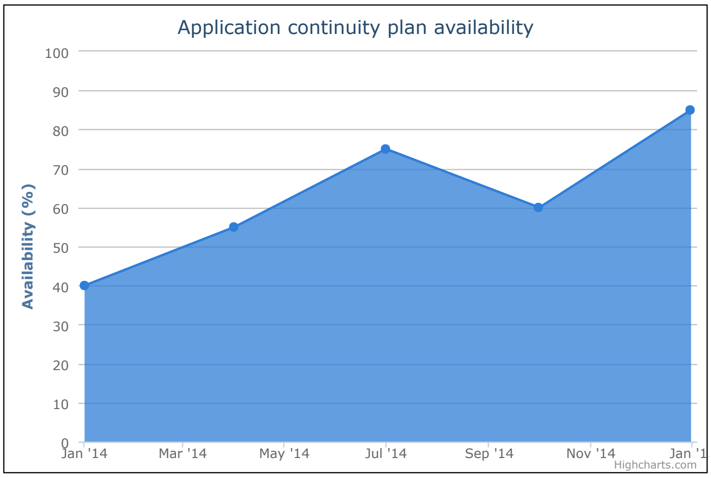


Figure 5.25.: Example of an area chart visualization according to [Sc14]

similar to line charts - they can visualize the changes of a metric value change over time.

Further, bar charts allow combined visualization of multiple metrics to support comparison as depicted in Figure 5.28. Nevertheless, in contrast to line charts, it is important that the y-axis is scaled linear and **never logarithmic**. According to Few [Fe09a], otherwise the meaning of the visualization will be distorted for the reader. For example, using a logarithmic scale with the base of 10, a bar encoding a value of 100 will be half as long as a bar encoding a value of 10000, cf. Few [Fe09b].

Pie charts A *pie chart* (cf. Figure 5.29) is a circular chart divided into sectors, whereas the arc length of each sector is proportional to the represented value. Usually, pie charts are utilized to represent ratios (cf. Section 2.2), i.e., the sum of the values of all sectors

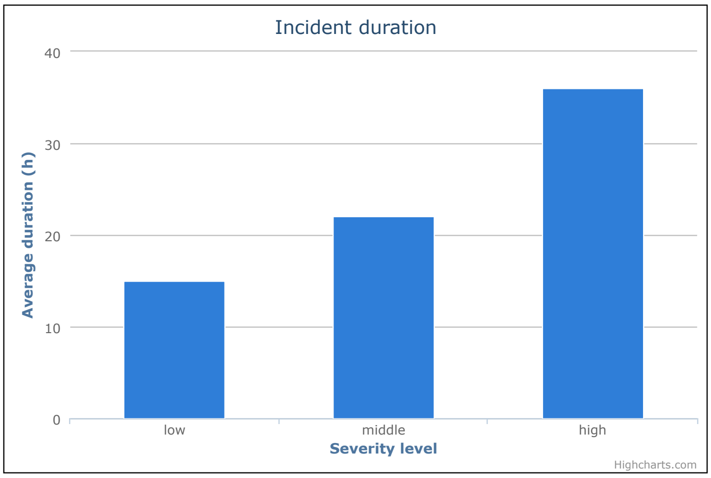


Figure 5.26.: Example of a column chart visualization according to [Sc14]

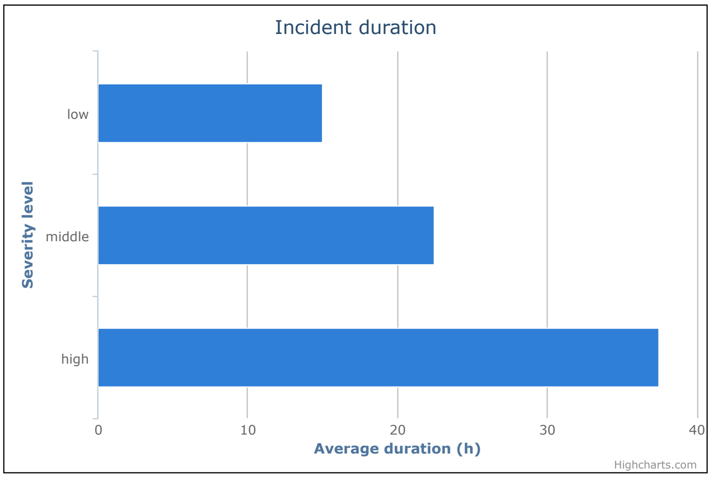


Figure 5.27.: Example of a bar chart visualization according to [Sc14]

equals 100%. Further, according to Cleveland [Cl85], pie charts are widely used in industry and the mass media, however many scientists **strongly disagree** with the use of pie charts due to the difficult decoding and comparison of values within pie charts (cf. [Cl85, Fe06, Fe07, Tu01]). According to Edward Tufte [Tu01], "[..]the only worse design than a pie chart is several of them, for then the viewer is asked to compare quantities located in spatial disarray both within and between pies". Therefore, Tufte [Tu01] and Few [Fe07] recommend to use tables or bar / column charts instead of pie charts, cf. Figure 5.30. Nevertheless, hence we aim to support typical metric visualization types, we allow the usage of pie chart in our prototype, nevertheless we recommend to our user to always critically reflect if a table or bar chart visualization is more suitable for their purposes.

Kiviat charts A *kiviat* or *spiderweb chart* (also named *radar chart* by some authors) is a radial visualization to display multivariate data, cf. Figure 5.31. Usually this type of visualization is utilized to compare objects on the basis of multiple comparison criteria.

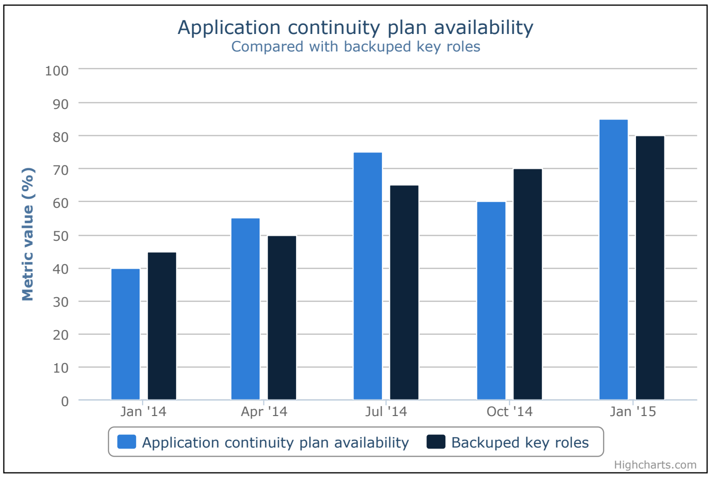


Figure 5.28.: Example of a column chart for the visualization of two metrics according to [Sc14]

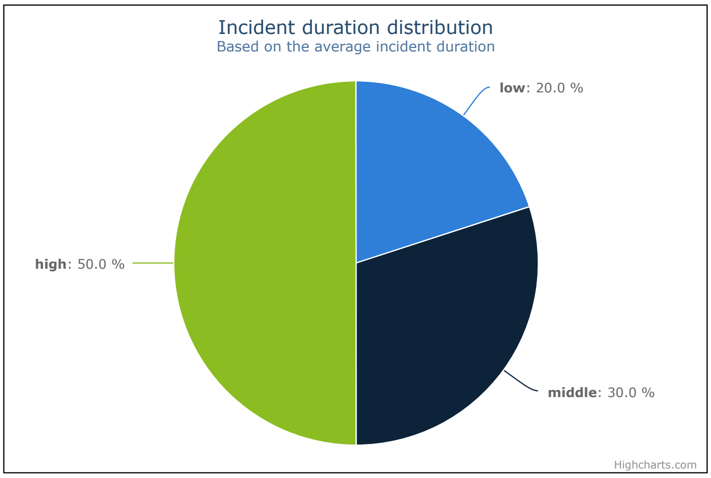


Figure 5.29.: Example of a pie chart visualization according to [Sc14]

Thereby, every criteria is represented with a spoke and the value for each measured object is marked with a dot. All dots belonging to one object are connected with straight lines (similar to line charts). Hence, kiviat charts can be understood as line charts transferred into a polar coordinate system.

For a kiviat chart, the x-axis has to consist of discrete categories, where every category corresponds to one spoke in the diagram. Although kiviat charts are often used in industry in dashboards (cf. [Bu06, Ke13, Ti08]), scientist criticize them as too hard to decode and often unnecessary, cf. [Fe09b, Ke13]. Usually the same data can be represented with a line or bar chart without losing information, cf. Figure 5.32.

According to Few [Fe05], the only three exceptions to use kiviat charts instead of line or bar charts are graphs having different quantitative scales, data fitting a circular display or graphs with the objective to represent symmetry instead of magnitude.

Bullet charts In business dashboards, so-called *gauge* visualizations as illustrated in Fig-

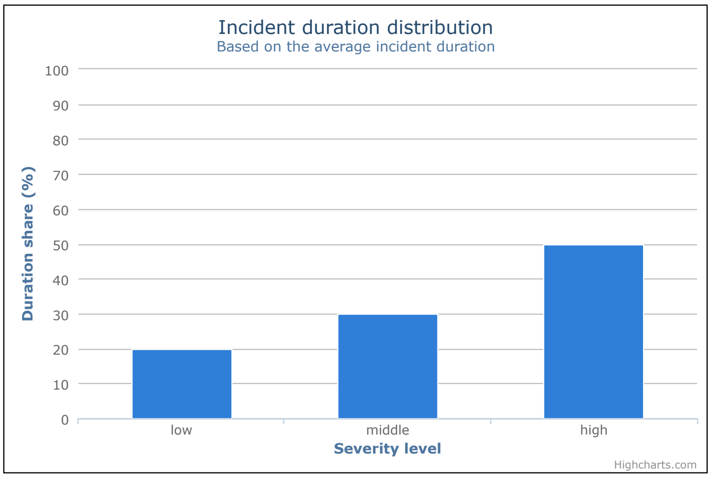


Figure 5.30.: Usage of a column chart instead of a pie chart (cf. Figure 5.29) according to [Sc14]

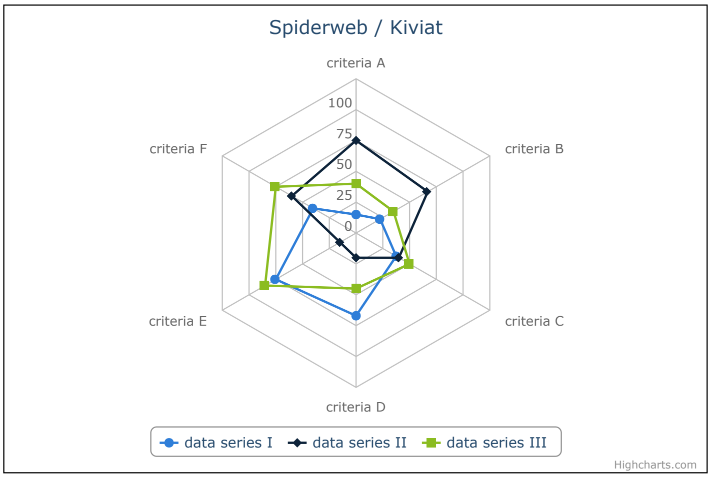


Figure 5.31.: Example of a kiviat chart visualization according to [Sc14]

ure 5.33 are often utilized to visualize the current snapshot value of a single metric. Although such visualizations are widely accepted in industry, researchers disagree with their usage hence they are hard to decode and compare (cf. [Ke13, Pa10]), and because they are "wasting" a lot of visualization space (cf. [Cl85, Tu01, Fe06]). To solve these problems, Stephen Few [Fe06] developed the so-called *bullet chart* as a solution for the visualization of the current value of a given metric, and in particular, for the usage of this type of visualizations within metric dashboards.

According to the author, (cf. Figure 5.34), a bullet chart consists of:

- a bar encoding the value of a given metric,
- a quantitative scale for easier decoding of the visualization values,
- background colors to encode qualitative ranges e.g. 'bad' (red), 'satisfactory' (yellow), and 'good' (green), and

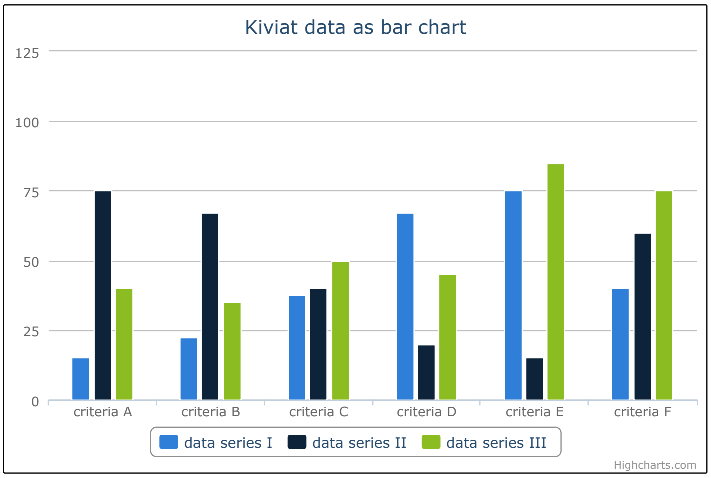


Figure 5.32.: Usage of a bar chart instead of a kiviat chart (cf. Figure 5.31) according to [Sc14]

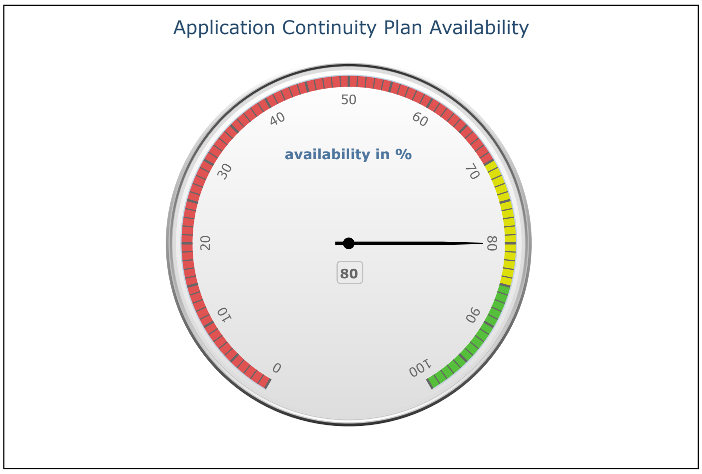


Figure 5.33.: Example of a gauge visualization for the metric *Application continuity plan availability* from the catalog according to [Sc14]

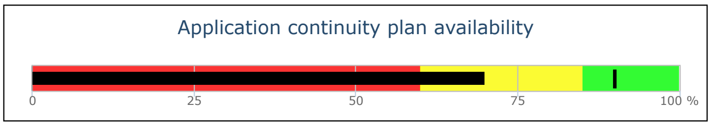


Figure 5.34.: Example of a colored bullet chart visualization according to [Sc14]

- a symbol marker to encode a comparative measure (in our terminology, this marker can be used to encode a target or planned metric value).

Nevertheless, the three colors used in the exemplary bullet chart visualization are not optimal from psychological viewpoint. According to the related literature, male humans can have problems in differentiating between green and red colors. Hence, even if these three colors are widely accepted in management disciplines, the literature suggest to use different gray color gradation, cf. Schätzlein' thesis for further details, cf. [Sc14].

Based on its linear design, bullet charts can be oriented either horizontally or vertically, and several charts can be combined in dashboards. For the visualization of EA management metrics, bullet charts can be utilized to provide snapshot views of metrics and corresponding planned or target value.

Accounting for these five types of recommended visualization for metrics, we performed an evaluation of existing web-based visualization libraries for the intended extension of Tricia allowing user-defined metric visualizations at runtime. For this purpose, we identified and compared four solution alternatives (*HTML5 Canvas*, *Raphaël*, *D3.js*, and *Highcharts*) as described in detail by Schätzlein [Sc14], resulting the selection of *Highcharts* for our intended software support. Further, Schätzlein describes in his thesis four distinct prototype stages, which played an important role during the solution implementation. The final prototype was integrated in Tricia and supports the user-defined visualization of metrics at runtime. For this purpose, the rich text editor of Tricia was extended by a corresponding visualization button (cf. Figure 5.35). By clicking the button, a visualization editor is opened, allowing users to

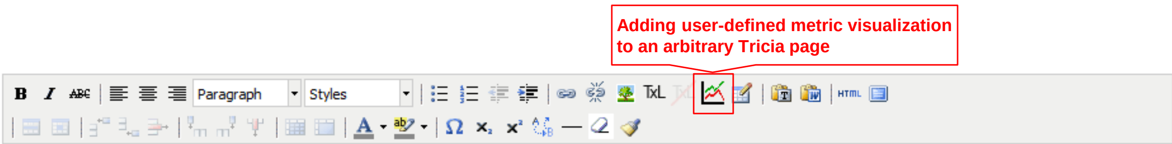


Figure 5.35.: Extension of Tricia's rich text editor by a button allowing user-defined metric visualizations

configure and embed visualizations within the rich text content of any Tricia page. Thereby, the configuration process of user-defined visualizations consist of the following three steps:

Selection of a visualization type As illustrated in Figure 5.36, a user firstly has to select one of the five supported metric visualization types. In this example, the user selects a bullet

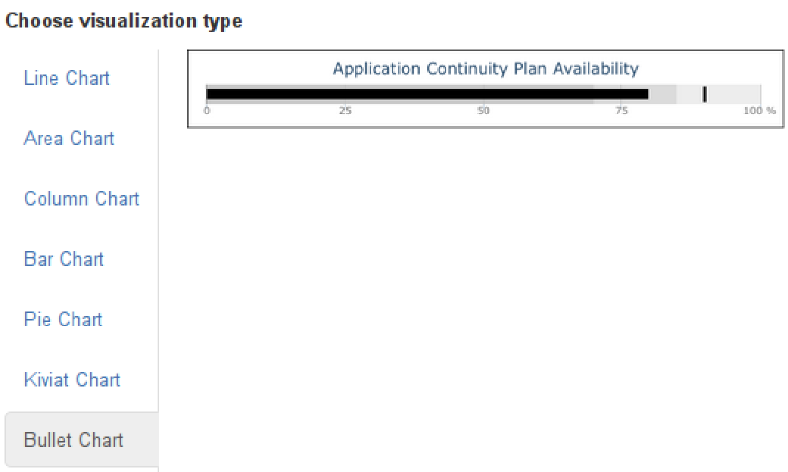


Figure 5.36.: Selecting metric visualization types in Tricia at runtime according to [Sc14]

chart. Further, to provide more guidance for the users during this visualization selection step, our prototype shows a simplified example of the selected visualization type in the upper part of the dialog (cf. Figure 5.36).

Selection of a corresponding MxL query Since every metric description has a corresponding MxL query (by the attribute formula of the type metric description), the user has to enter the name of the custom function implementing the calculation rule of a given metric, cf. Figure 5.37. For bullet charts, users can additionally define a target value.

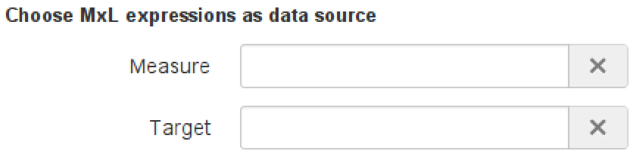


Figure 5.37.: Entering the name and target value of a custom function to define a corresponding visualization at runtime according to [Sc14]

Configuration of the visualization As depicted in Figure 5.38, in the third step of the process, the user has to fill out a specific configuration for the selected visualization. Thereby, the

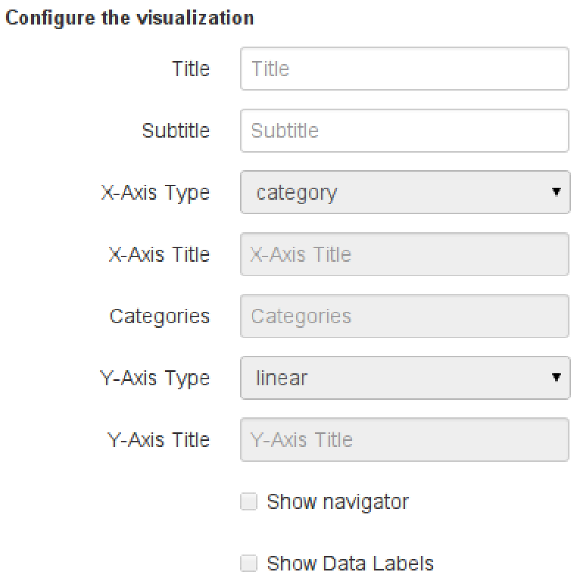


Figure 5.38.: Adjusting specific visualization properties for a selected metric visualization at runtime according to [Sc14]

configuration values depend on the type of the selected visualization (cf. Schätzlein [Sc14] for additional information). In our example (a bullet chart), the user can enter a *title* and a *subtitle*. Additionally, the user can select by clicking the corresponding check boxes in the bottom of the view, whether the visualization should show data labels and whether a navigation frame supporting zooming operations within the visualization at runtime should be shown.

Further, Figure 5.39 provides a detailed overview of the mapping of configuration properties for the visualization of the catalog's metric *Application continuity plan availability* as a line chart.

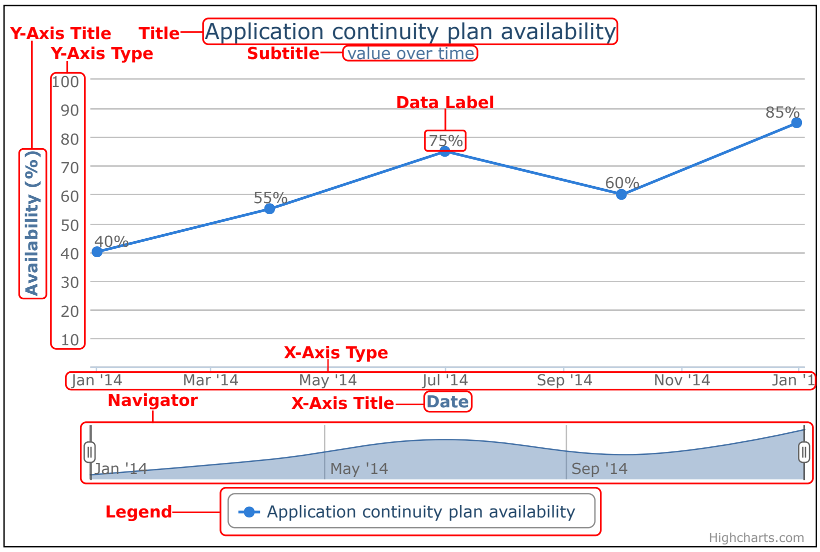


Figure 5.39.: Mapping of configuration properties to a line chart visualization according to [Sc14]

Consequently, our prototype supports user-defined metric visualizations. Further, by visualizing MxL expression using the @ operator, our prototype supports the visualization of metrics accounting for the historicization of the underlying (qualitative) EA model. For more informations regarding the design and implementation of the metric visualizations, we refer to Schätzlein's thesis [Sc14]. Please note that all screen shots provided in this Subsection were done in our prototype by using corresponding test data.

5.4.5. Web-based Editor Support for Expressions

To support Tricia users in the management of TxL 2.0 expressions, we support defining and embedding expressions in the rich-text content of arbitrary Tricia pages. As illustrated in Figure 5.40, we extended the page editor by a corresponding button, which opens our expression editor. Figure 5.41 shows an example of writing MxL code. Thereby, our prototype provides following support:

Syntax highlighting The editor supports syntactic highlighting by coloring (e.g. keywords, and strings) In this example, the *find* keyword is colored blue, whereas the string *Business Application* is colored gray.

Code completion The editor provides a list of possible identifiers based on a prefix as defined by the user. Thereby, the code completion is triggered by simultaneously pressing the keys *CTRL+SPACE*. The list of proposals contains elements from both—TxL 2.0 elements (e.g. existing custom functions) and Tricia model elements (quantitative EA model elements), e.g. existing types and attributes. In this example, the type *Functional*

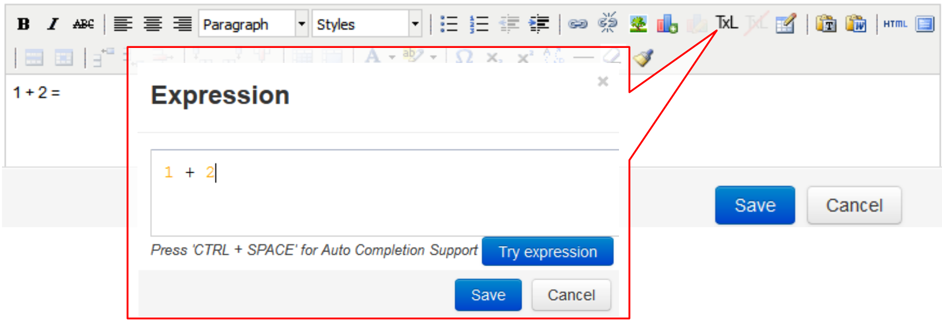


Figure 5.40.: Calling the MxL code editor from Tricia's rich-text editor according to [Re13]

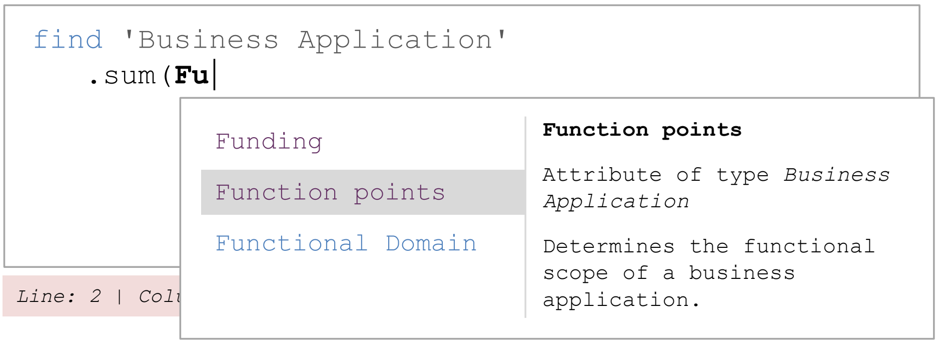


Figure 5.41.: Writing a TxL 2.0 expression at runtime according to [RMM14]

Domain as well as its two attributes *Function points* and *Funding* are suggested. Hence no existing TxL element starts with the prefix 'Fun', no corresponding proposal for TxL elements is offered.

Integrated documentation The editor in addition displays the documentation of a marked proposal in the code completion list. In the given example, a user has selected the proposal *Function points*. Therefore, the editor loads the documentation of this model element and shows the retrieved documentation to support the user in quickly deciding, if this proposal is suitable.

Error localization The editor highlights the origin of syntactic or semantic errors in the entered MxL code. If a syntactic error (e.g. missing closing brackets), or a semantic error (e.g. unknown type) is determined during the evaluation of the expression, the editor displays the position (by showing the corresponding line and column) of the error's origin. For example, if the type Business Application is not defined in Tricia at the time point of the evaluation of the expression, the editor will show a corresponding error message referring to line 1 and column 5.

Testing expressions The editor support the direct test of expressions and hence, users can save efforts in saving the expression and calling it from a different location. An expression test can be triggered by simply clicking the button *Try expression*, cf. Figure 5.40.

5.4.6. Implementation of the MMFS in Tricia

As discussed previously, we implement our MMFS structure as a Tricia type called **metric description**. All of the MMFS elements are implemented by corresponding attributes. Further, for each instance of a metric description, the newly introduced *formula* attribute allows the link to a corresponding custom function implementing the metric calculation rule. Although this implementation of the MMFS as a type in Tricia is intuitive, the requirement that the layout of the MMFS is displayed in Tricia, proved hard to solve. Although Tricia supports a HTML-based template engine, there is just one template affecting all pages without taking into account the type definition assigned to a page. More precisely, only the appearance of the so-called *built-in* attributes is adjustable by the page template, e.g. the name *name* and the rich-text content of the page, nevertheless, all type-specific attributes (e.g. the attributes *Salary* and *Hours* of the type *Employee* in Figure 5.19) are excluded. Hence, there are no restrictions regarding the page's layout and design.

To enable type-based templates, Tricia has to be extended to apply layout templates on each of the information model's types in order to apply the layout on all type instances. Furthermore, since the template has to be HTML based, the definition of type-based *Cascading Style Sheet* (CSS) classes allows a proper organization of the design of the type's instances. By

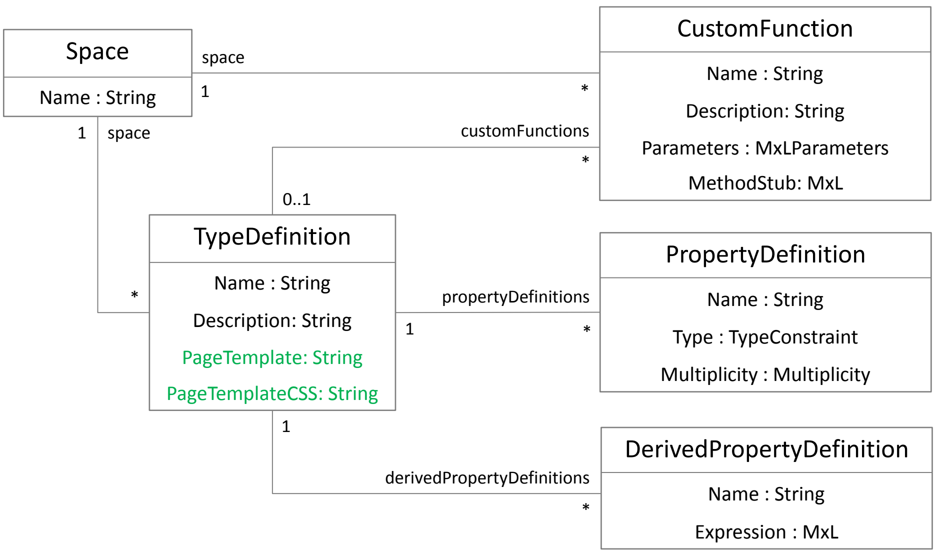


Figure 5.42.: Extension of Tricia's *TypeDefinition* by the properties *PageTemplate* and *PageTemplateCSS* according to [Re13]

implementing a type-based template engine in Tricia, the type definition is extended by the two properties *PageTemplate* and *PageTemplateCSS* defining the appearance of pages which are assigned to this type (cf. Figure 5.42).

For the definition of the template and the CSS classes a new view *Page Template* for type definitions has been implemented. Initially, both—the template and the CSS classes are empty, as shown in Figure 5.43. Since the template is internally managed as a rich-text property, it is edited via Tricia's *TinyMCE*-based rich-text editor. However, in contrast to rich-text contents

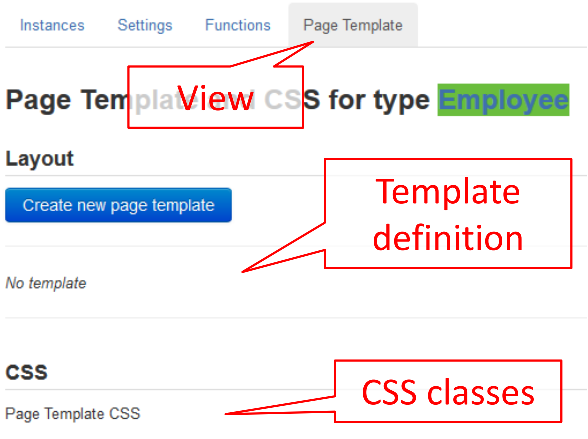


Figure 5.43.: The *Page Template* view of the exemplary type definition *Employee*, whereas neither a template nor CSS classes are defined according to [RMM14]

of page instances, page templates allow the embedding of properties, i.e., they position the corresponding type attributes. For this purpose, the rich-text editor provides a button for embedding properties to define page templates. This button opens a *modal* dialog containing a list of all currently available type attributes (*built-in*, *regular*, and *derived attributes*) Thereby, build-in attributes are colored black, whereas regular attributes are colored green and derived attributes are orange. An additional drop-down list allows to select, whether the attribute's name, value, or type has to be inserted within the template (cf. Figure 5.44). By clicking the *save-button* of this model dialog, the selected options are serialized to a JSON-object and are stored in Tricia. For more details regarding the implementation of the template engine in TxL 2.0, we refer to the work of Reschenhofer [Re13].

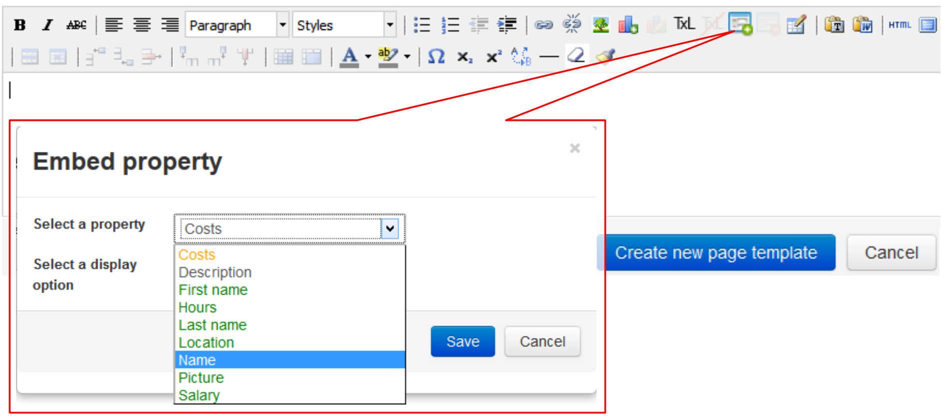


Figure 5.44.: *Add/Edit property* button for defining page templates according to [RMM14]

To implement the desired type-based layout for *metric description* instances, we defined a corresponding page template (cf. Figure 5.45), whereas the layout is aligned with the proposed graphical organization of the MMFS structure, cf. Section 3.4. This template is applied on each of the type's instances. Figure 5.46 shows an example of the application of this

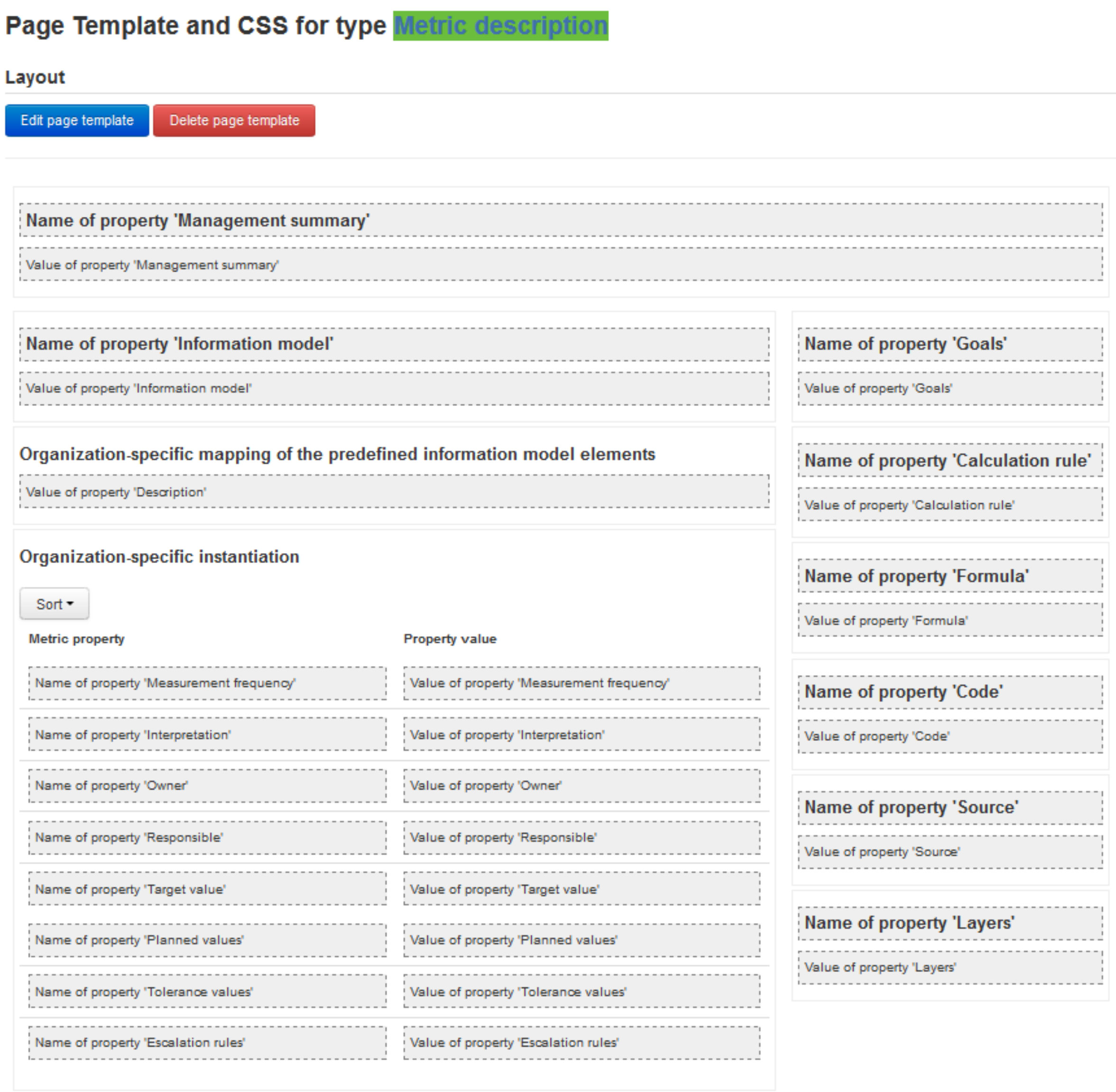


Figure 5.45.: Template definition for the type *metric description* according to [RMM14]

type, when the page representing the metric description instance *Application continuity plan availability* [Ma12a, p. 19] is rendered in the browser. Further, Figure 5.47 provides an example of the implementation of this metric from the Tricia version of our catalog. All other metrics, as well as their MxL implementations, are available on the pages of our chair [Se14]. The proposed implementation supports further the organization-specific extension of the MMFS implementation. For instance, according to the results of the evaluation of our method, an organization may decide to extend the MMFS structure by the element *Organization-specific goal description*, cf. Section 4.1.2. For this purpose, the organization needs firstly to extend the type *metric description* by a corresponding attribute. Secondly, the type-based template should be correspondingly extended. Thereafter, whenever an instance of the type *metric description* is rendered, this additional description element is shown.

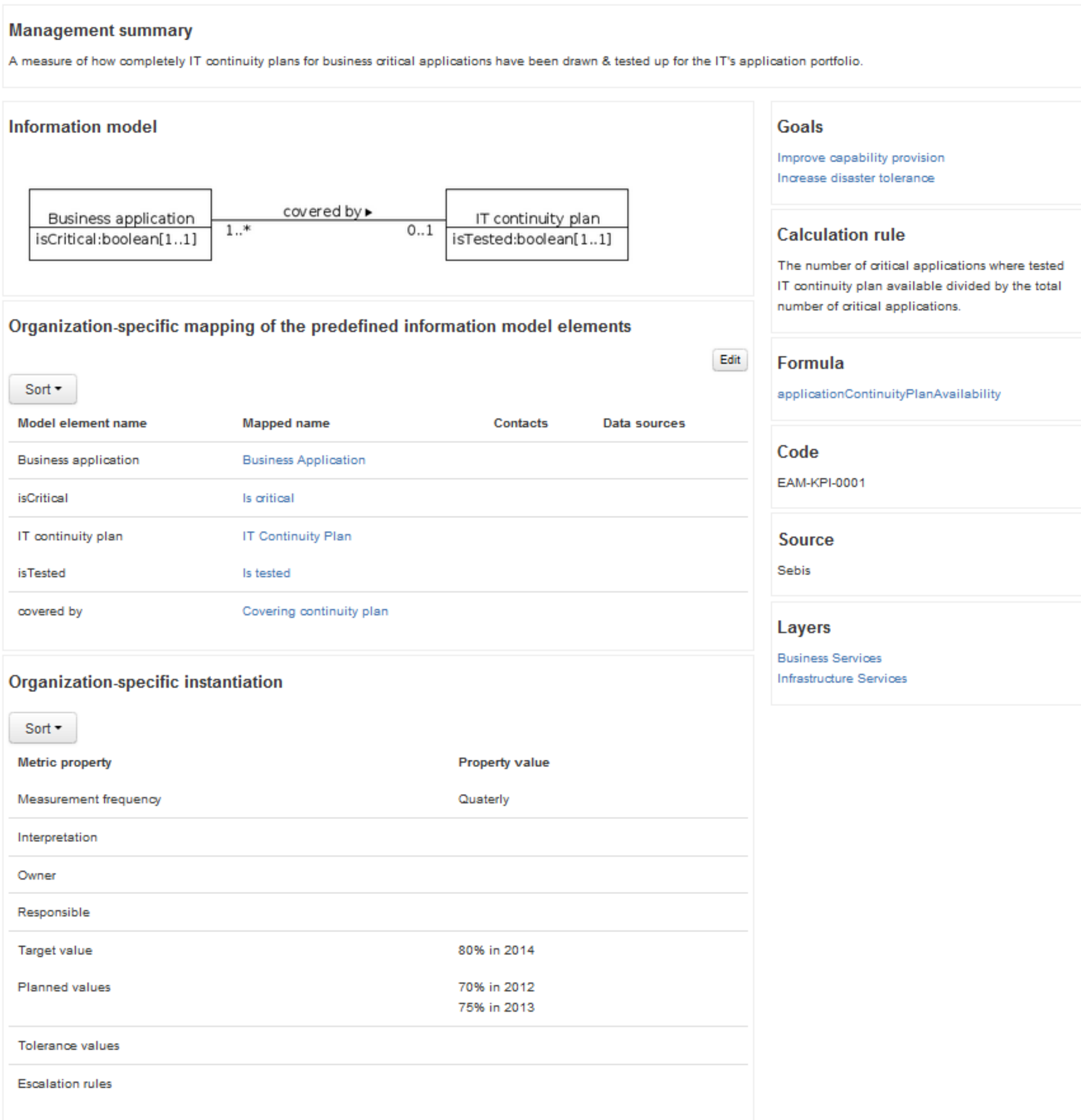


Figure 5.46.: Representation of the metric *Application continuity plan availability* based on the layout definition for the type *metric description* according to [RMM14]

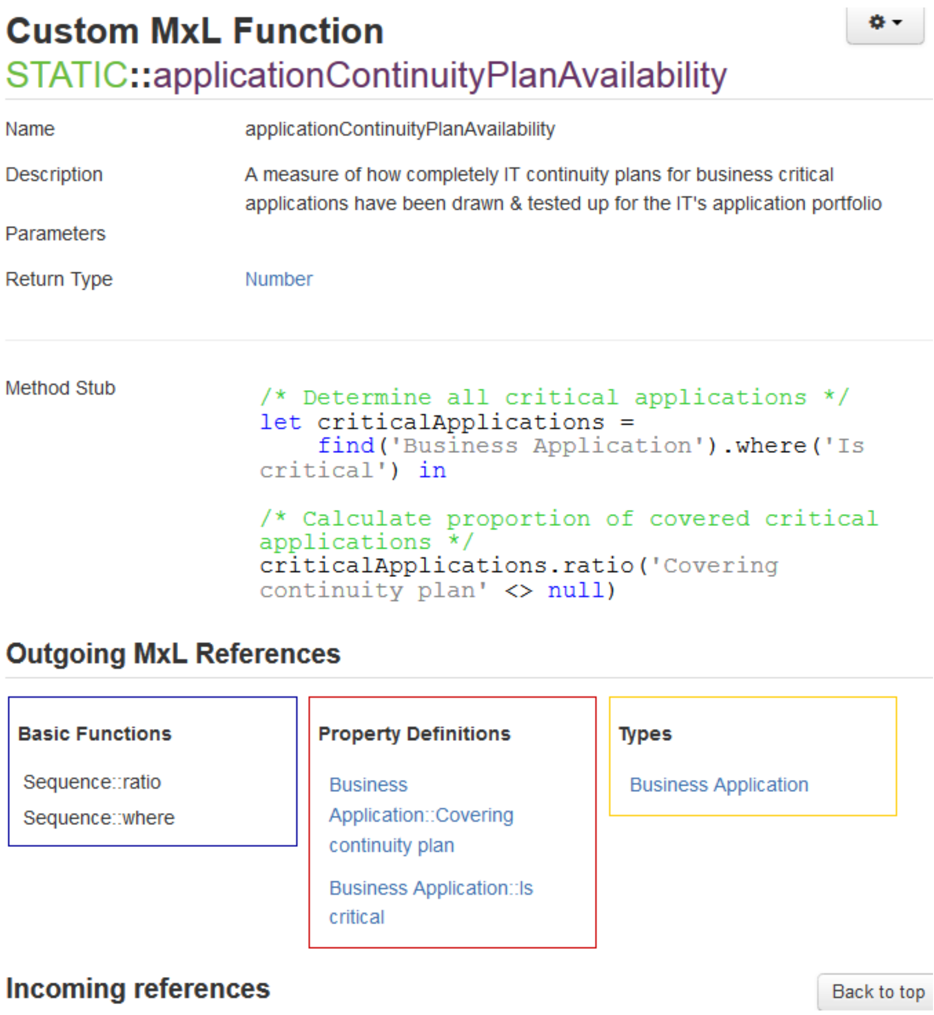


Figure 5.47.: MxL implementation from the Tricia version of our catalog

CHAPTER 6

Evaluation

According to our research method (cf. Section 1.2), the evaluation of the design and the demonstration of the utility of the created research artifact (the presented software prototype) is an essential part of the research process (cf. Hevner et al. [He04]).

To appropriately evaluate our prototype, we aimed at conducting a real-life case study of the application of our solution. For this purpose, we asked those experts, who are known to us in terms of employing EA management metrics, for their interest in the application of our prototype in their organizations. As previously mentioned in Section 5.2.1, one of the experts of the $CALM^3$ workshop, who represents a mid-sized and internationally operating German bank, agreed with the prototypical usage of Tricia. The expert in particular showed interest in the adoption of suitable metrics from our catalog for his organization and in addition, in the comparison of our prototype with the currently employed self-developed solution used by his organization. Thereby, the bank uses a commercial tool for the management of qualitative EA management models and a *business intelligence* (BI)-based solution, in which specific parts of the EA model and its instance data are imported to calculate a set of predefined (confidential) EA management metrics. These metrics are used by the enterprise architects and their management for the EA planning and optimization purposes.

6.1. Structure of the Case Study

Having the opportunity to conduct only one case study for the application of our prototype, we decided to perform a qualitative evaluation of our prototype based on the subjective feedback of enterprise architects from the bank who personally used our software. Figure 6.1 illustrates the structure of the case study. Thereby, we firstly conducted a 2.5 hours' workshop with two enterprise architects from the bank, concerned with the introduction to our conceptual metric management method in detail, as well as an introduction to the usage of the prototype (e.g.

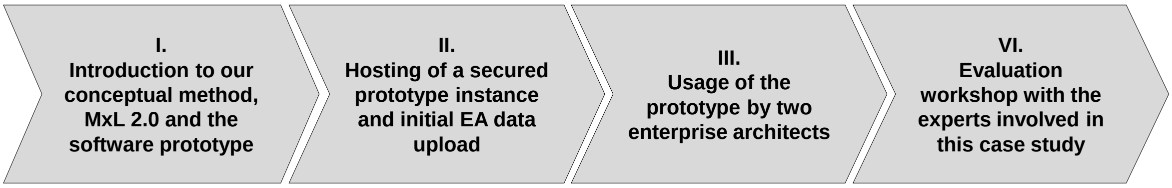


Figure 6.1.: Structure of the case study

introducing the implementation of the MMFS structure as a type, the implementation of the catalog as a workspace, as well as an introduction to MxL and the different types of MxL providers and their application).

After this introductory workshop, we deployed a dedicated Tricia instance on a secured virtual machine, hosted by our chair to ensure access only to selected members of our chair and the two participating enterprise architects from the bank. Additionally, the experts provided us a set of anonymized qualitative EA model data as a base for the prototypical application of our solution. We imported this EA data by using Tricia's Excel import feature into a cloned workspace of the catalog, according to our solution design. After performing this experiment preparation, we notified the experts that the prototype is prepared and can be used for the given purpose.

The experts started to work independently with the prototype. In the beginning, they reported several login problems, which turned out to be an issue based on the cookie management of their browsers (Internet Explorer v.6). After deleting all cookies, the experts have been able to successfully login for the rest of the time. As one of the experts reported, he used the prototype also from his private computer in the evenings, where he used Firefox and Chrome browsers and he has not experienced any login problems. After approximately two weeks of independently working with the prototype, the experts contact us to share the results and to give feedback in a corresponding evaluation workshop. Consequently, we scheduled and conducted a 2.5 hours' evaluation workshop on May 16^{th}, 2014.

6.2. Expert Interview Design and Results

Since we have only one case-study and two experts as interview partners for the evaluation of the prototype, we decide to focus on the collection of qualitative and subjective feedback in the evaluation workshop. For this purpose, we define the following antithesis with respect to our research hypothesis (cf. Section 1.2):

> ***Research antithesis:*** An organization with an established EA management function cannot define and implement a holistic life-cycle management of organization-specific metrics using our software supported method.

With respect to this antithesis and in line with the decision to collect subjective and qualitative experts' feedback, we have consequently defined a list of 50 open questions, targeting the validity and appropriateness of our architectural and implementation decisions as well as the documentation of the experts' assessment of the prototype. The complete questions list is documented in the Section A.

Further, we have structured the workshop in three parts as follows:

MxL design - This part is concerned with the validity and completeness of all requirements for the MxL design. In addition, we collect subjective qualitative feedback regarding the general MxL design based on the experts experience with the prototype (cf. Section 6.2.1).

TxL and Tricia extensions This part of the workshop is concerned with the validity and appropriateness of our implementation decisions made during the development of the software prototype, in particular the introduced extensions of Tricia, e.g. the MxL query editor and the metric visualization component. In addition, we collect subjective qualitative feedback regarding these features, and we focus on the identification of implementation shortcomings and missing features (cf. Section 6.2.2).

Support of a holistic metric life-cycle management method This part is concerned with validity and appropriateness of the design decisions regarding the implementation of our conceptual solution as well as all related artifacts, e.g. the proposed implementation of the MMFS structure and the catalog. In addition, we document the user experience and opinion towards the degree of provided software support for our conceptual solution (cf. Section 6.2.3).

Here, we want to provide more background information of the experts involved in the case study. The first expert is currently employed as enterprise architect and has been working for this bank since more than 17 years. The second expert has more than five years of working experience in the bank, in particular in the domains BI and data analyses. The expert joined the EA management team in the beginning of 2014, where she already gained basic experience with the EA of the organization, as well as with the BI-based metric management solution used by the bank.

Further, to ensure a proper transcript of this 2.5 hours long workshop, we got the permission to record the talk. Therefore, we are able to present many citations from the interview in the subsequent sections. Please note, that we translated these statements from German, without changing any given assessments or opinions. Further, we anonymized the identity of the interviewees according to their personal wish. In addition, after transcribing and translating the interviews, we deleted the recordings of the talk as we were asked to in advance.

6.2.1. MxL 2.0 Design Evaluation

In this part of the interview, we asked the experts about their opinion regarding the validity and appropriateness of our requirements for the MxL design (cf. Section 5.1.4). Thereby, both experts confirmed the validity of the requirement for a minimal language design, ensuring a sufficient expressiveness for the calculation of the catalog's metrics. One of the experts said: "I like this minimality approach, hence I am not overloaded with features from a user's perspective". Asked about the completeness of MxL's expressiveness, both experts confirmed they were able to find related language constructs for all their purposes. Hence, the experts have not been able to name concrete missing MxL operators or constructs. Nevertheless, one reported problems during the learning of the language's syntax. The expert stated "I spend approximately 30 minutes to learn and understand how to get a specific element from a MxL map". Although the experts agreed on the completeness of the provided language specification (this documentation was available as a separate workspace in the used Tricia instance, where the experts had read access), both of them strongly recommended the extension of the current MxL documentation by a so-called *beginner's guide*, teaching the basic MxL concepts to improve the learning process of the language from programmer's perspective.

Further, both experts confirmed the validity of the requirements to design MxL as a functional language and in particular with the support of (implicit) lambda expressions and higher-order functions. Asked about their experience with these functional language features, both experts confirmed to have used lambda expressions during the implementation of their metrics, as well as a simple recursion. Thereby, based on the hierarchical structure of functional domains in their EA (a functional domain can have an arbitrary number of sub-domains) the experts have defined a MxL expression calculating the number of specific business applications per functional domain. Nevertheless, both experts clearly said they did not use any further higher-order functions. Although, both of them agreed, that this feature will be frequently used, if the prototype is used productively by their organization. In addition, both experts reported they have needed a couple of hours to understand these functional features, since none of them is experienced with functional programming language. In this context, one of the experts stated: "The last time I used a functional language was in the university, approximately 30 years ago, when I took a corresponding computer science course." Nevertheless, the experts reported to have liked these functional features after understanding the MxL basics, allowing them to easily extend simple expressions by concatenating additional "language building blocks" to implement more complex logic. One of the expert stated: "You need first to rethink and to understand these paradigms. However, once you have understood the basics, it makes really fun to use this language, since you can achieve a lot by writing only few lines of code. Even more, as I started to feel familiar with the language, I did not wanted to stop coding."

Further, both experts confirmed the validity of the requirement to allow the access to the underlying qualitative EA management history in MxL expressions. As the provided test data did not contain history data, the experts were not able to test this prototype feature and thus, they could not provide relevant feedback regarding the implementation. Nevertheless, one of the experts said: "It is essential to allow the excess to EA model history data, as enterprise architects want to understand the chronological development of their metrics based on changes in the underlying EA model. Therefore, I fully agree with the decision to allow EA model history access in your language". Further, the experts described a different approach of history management. Thereby, the bank models for instance planned introduction and shut-down dates for business application explicitly in their EA model. In this context, the experts agreed, that our prototype supports also this type of history modeling by filtering the corresponding date attributes with existing MxL constructs. Nevertheless, the experts proposed to extend the list of predefined find-operators, where a date range can be defined by system users in the browser, for instance to support reporting of planned application introductions in quarter 3, 2017 by adjusting a corresponding search parameter.

Both experts further confirmed the validity of the requirement to design MxL as a (static) type-safe language. One of them further stated: "Generally, I prefer static type-safety more than dynamical typing, since this paradigm helps to avoid typing errors and provides more comfort from a programmer's perspective. In addition, the validity of expressions in type-safe languages can be checked at compile-time, whereas dynamically typed languages can be checked first at run-time."

Surprisingly, the experts strongly agreed with the validity of the requirement to design MxL also as an object-oriented language. One of them even stated: "This design decision is highly convincing in my opinion. For me, the provided opportunity to attach MxL expressions to any type in my EA model is a highly elegant solution for the purpose of using metrics in our domain. I cannot image how the language would look like, if it was designed only as a functional one. Further, in my opinion, the mixture between object-oriented and functional language design is very useful for the given problem, in particular, I very much like the way how one can work with all types of EA objects in the browser. For me, the benefit of this mixture of language paradigms is that one can simply select an arbitrary EA model type and immediately define required function in the browser." In addition, the other expert reported she liked the common usage of the '.' operator, which she knows from other object-oriented programming languages.

Both experts further agreed with the validity of the requirement to support recursion in MxL. Although the experts used this feature only prototypically, they explained the importance of this functionality with concrete examples from their EA model, where several hierarchical structures are defined. Correspondingly, quantitative models implemented in MxL have to account for the correct computation of these hierarchical structures by allowing transitive calculations.

In addition, both experts confirmed the validity of the requirement to support backward navigation in MxL. One of them stated, that this features positively affects the syntax of the language, hence the *whereis* operator prevents the writing of additional sub-queries as typically done in SQL. Nevertheless, the expert criticized the documentation of this feature, as concrete examples for the syntax of this operator are missing.

Asked for their general feedback and experience with the MxL design, the experts reported they had been able to implement all their metrics using the provided MxL constructs. One of them further stated: "According to my experience with your prototype, MxL seems to provide a sufficient support for the implementation of metrics in our domain. For our cases, we could always find appropriate language features for the implementation of our metrics. Nevertheless, it will be interesting to see, if a bigger set of metrics, will indicate shortcomings in the expressiveness of your language. However, currently, we do not see any concrete requirements, which are not covered by your language design." Additionally, both experts emphasized one more time the importance of providing a beginner's MxL programming guide to allow quick and guided learning of the language besides the existing language specification.

6.2.2. Evaluation of TxL 2.0 and the Related Tricia Extensions

In this part of the interview, we focused on the evaluation of the validity and appropriateness of the design decision to perform metric management within an EA management tool and to integrate MxL in Tricia for this purpose. Hence, asked about their opinion regarding this design decision, both experts strongly agreed with the appropriateness of this idea. One of them even stated: "This integration is a huge benefit. Your solution is much nicer and flexible compared to my solution for the given problem, where I use work-intensive processes to import and prepare the data for the calculation of our metrics. Further, if you decide to use a BI-based solution, you have to firstly make several time-consuming design decisions for your reports. In this context, you need to define specific dimensions and fact tables, than you need to import your data, and afterwards you have to define and implement your metrics and correspondingly to define a table, a column and an attribute required for the calculation of a metric. With your solution, I just need to define a new MxL expression and to perform three mouse clicks to archive the same results, especially for rudimental metrics consisting of up to five lines of code. Your solutions makes definitely more fun to use and is clearly faster then our own BI-based one, where we need definitely higher working efforts."

Further, one of the experts emphasized the benefit provided by the integration of MxL in Tricia in terms of allowing emergent quantitative modeling capabilities. The expert reported he firstly started with the definition of derived attributes within specific EA types, "where they intuitively belong to". In this manner, the prototype supported the expert to quickly define several intended quantitative models as derived attributes. At a given point in time, the expert realized, that he does not require so much derived attributes, thus he performed a refactoring of his code by defining a type function, i.e., a function attached to the given type. Thereby, this function combined several of the quantitative models previously expressed by multiple derived attributes. Further, the expert identified specific type functions, which he redefined as MxL custom functions in terms of concrete EA management metrics. According to the expert, in this way, organization-specific metrics emerge naturally, which represents a significant benefit in his understanding towards the definition and management of metrics.

Additionally, both experts confirmed a need for metric visualizations and stated, they plan to use this feature in future, although they have not tested this feature yet. Nevertheless, to our surprise, the expert reported that metric visualizations are not the preferred communication of results to their upper management. Instead of visualizations, their management prefers results expressed in monetary units. However, the experts reported that the supported metric visualizations are considered as useful for discussions within the EA management team and for the communication with IT employees. In this context, one of the experts stated that metric results could be incorporated within existing software maps, e.g. process-support-maps. This idea is in line with the metric visualization concepts described by Lankes [La08].

Both experts further confirmed the validity of the design decision to support metric relationship management by the prototype. One of them stated in this context: "This functionality is in particular important for a productive system. Further, the importance of this feature becomes more critical over time. After using this system for several years and when the involved people start to change, this mechanism is the only way to support the users to understand the effects of an attended change of a given MxL expression on other existing MxL functions. The way how you have implemented these relationships in your MxL expressions view is very well-known to me from several existing software development tools. For me, this solution looks cool." In addition, the experts reported they consider the general idea of metric relationships management as helpful especially for metrics, which are developed by other users and for the purpose of refactoring metrics. In this context, one of the experts reported he very much liked the feature of the prototype, where renamed functions are automatically updated in affected MxL providers - "a feature, which is usually supported by software development tools".

The last part of this interview section was concerned with the evaluation of the provided web-based MxL development user support. Thereby, right after the moderation of this questions block, one of the participants stated: “It is really great, that you can simply open one window to define a derived attribute and to enter the code directly there. Even more, it is very helpful, that you can evaluate and test your code immediately while you are writing it. In my opinion, this feature is worth a mint, in particular when you are new to the system and you are learning the MxL syntax.”

Further, both experts confirmed the provided benefit regarding better code understanding from programmer’s perspective by the syntax highlighting feature. One of them said: “Syntax highlighting is important, since you are writing programming code. This feature is always helpful in the domain of software development.” Further, the other expert said: “The only minor issue we experienced with the code editor is the highlighting of brackets. Thereby, when you write an opening and closing bracket after each other, the graphical representation in green color of these brackets is somehow confusing, in particular for a person, who looks at the monitor from the side during pair programming activities.” Further, one of the experts recommended to include a so-called *code-snippet* feature in the prototype. According to the expert: “[...] it would be great to allow programmer-defined ‘code snippets’. Thereby, MxL developers could store frequently used code pieces within a sandbox. In this manner, the programmer can easily copy-paste these code snippets whenever required.”

In addition, both experts assessed the auto-completion feature of the prototype as helpful in general. Nevertheless, surprisingly, both reported to have experienced irritations with the high number of provided suggestions by this feature. As example, one of the experts stated: “If you just write the ‘.’ operator and then trigger the auto-completion, you get a list of all existing functions and attributes, which is irritating, especially when you are new to the system. In addition, if you have a large EA model, the current implementation of this feature will show for example all existing attributes. Hence, we want to encourage you to minimize the number of shown suggestion, or at least to define specific suggestion categories, e.g. type suggestions, attribute suggestions, or basic MxL function suggestions.”

Furthermore, both experts fully agreed with the benefit provided by the implemented MxL error message engine. One of the experts reported, she experienced a lot of syntactical problems while learning MxL. According to the expert, the provided error messages, in particular the indication of the exact problem location in the code, was very helpful to quickly understand what is wrong and how to solve the compile problem.

6.2.3. Evaluation of the Provided Software Support for our Holistic Metric Life-cycle Management Method

In this part of the interview, we focused on the evaluation of the validity and appropriateness of the implementation of our conceptual artifacts, as well as the evaluation of the provided degree of software support with respect to our conceptual solution based on the experts’ experience.

Both experts fully agreed with the validity of the decisions to implement the MMFS structure as a type (*metric description*), as well as with the implementation of the catalog as a separate workspace. Further, the experts confirmed the validity of the decision to extend the MMFS implementation by the additional attribute formula, allowing the formal definition of metric calculation rules as MxL functions. In this context, one of the experts stated: “By the combination of the underlying EA data, the related metrics as MxL functions, and the full metric documentation in one tool, your solution is very helpful to perform efficient and intuitive management of metrics. Further, I can keep an overview over my metric system and in addition, I can quickly perform required changes to this system. This is a huge benefit compared to our own solution, where the metrics documentation is not contained in the tool, but stored in a separate Office document. This quickly leads to the problem of obsolete metric documentation. In your solution, you need only one mouse click to jump from the documentation to the implementation of a given metric."

Further, both experts strongly agreed with the validity of the decision to allow user-defined layout definition for Tricia types at run-time. The experts reported, they can imagine to use this feature for all of their types after discussing the required layout with their colleagues and partners from both - business units and IT departments.

Additionally, both experts agreed with the validity of the decision to implement the catalog as a wiki and to initialize the metric life-cycle management method with a clone of the catalog. In this context, the experts reported that the provided navigation support (e.g. the Goal-Concern-Matrix) is appropriate and helpful to quickly identify related metrics. Nevertheless, to our surprise, one of the experts reported, he intentionally did not use the provided navigational structures. More precisely, the expert said: "I was interested very much in understanding all of your metrics, hence I decided to spend more time and to study all of them in detail. Thereby, I discovered several very interesting metrics, which I selected to use for my EA management function. Nevertheless, if your catalog would contain several hundred metrics, I will use your navigation structure, since it is not realistic to study that many metrics in detail."

Moreover, the experts described the process of selecting, adapting and defining metrics as well-supported by the prototype. Thereby, the experts showed several examples for each of the corresponding sub-steps of our metric management method, whereby they have selected, adapted and deleted metrics from the catalog clone. In this context, the experts showed even organization-specific instantiations of these metrics, where the corresponding concepts from their EA were linked to the general suggestions provided by the *metric description* instances. In addition, the experts have defined exemplary (confidential) metrics, which the bank already manages in their own software solution. For the purpose of the evaluation of this thesis, the experts allowed us to present one of these metrics and to include it within the catalog. The MMFS description of this new metric is depicted in Figure 6.2. Thereby, according to the experts, several business units employ self-developed Office-based solutions (e.g. complex MS Excel spread sheets, or MS Access solutions) as IT support for specific (in some cases even critical) business processes. All of these applications are known and documented in the EA of the bank, since these applications are subject of audits by regulatory bodies. From an EA management point of view, such IT solutions for (critical) business processes are an indication of insufficient IT support, since these applications are not covered by the release, change and disaster management functions of the IT organization. Hence, the enterprise architects are interested in calculating and minimizing the number of such software solutions.

Further, the experts reported three issues they have experienced during their work with the prototype, which have to be improved by a next versions of the software:

1. The majority of the *metric description* type's attributes were defined as *required attributes* in the prototype. Therefore, whenever a new metric description instance is

Ratio of unsupported MS Office-based applications by the IT

Management summary

Specific business applications are MS Office-based business applications developed by business units and hence not supported by the IT. This is considered as insufficient IT support for the business units.

Information model

Business Application
name:string [1..1] isMSOfficeBased:boolean [1..1] isITSupported:boolean [1..1]

Goals

Ensure compliance
Foster innovation
Improve capability provision
Improve project execution
Increase disaster tolerance
Increase homogeneity
Increase management satisfaction
Increase transparency
Reduce operating cost
Reduce security breaches

Organization-specific instantiation

Mapping:

Name in model	Mapped name	Data contact	Data source
Business Application			
name			
isMSOfficeBased			
isITSupported			

Properties:

Metric property	Property value	Observed values
Measurement frequency		
Interpretation		
Owner		
Responsible		
Target value		
Planned value(s)		
Tolerance value(s)		
Escalation rule		

Calculation rule

Number of MS office-based business applications not supported by the IT divided by the total number of business applications.

Code

EAM-KPI-0084

Source

Industry partner

Layers

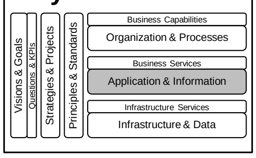


Figure 6.2.: MMFS description of the observed new EA management metric in the case study

created, i.e., a new metric is defined, the user has to immediately provide all attribute values, which is however complicated, especially if the image representing the information model is yet not been uploaded to the prototype. To solve the issues, the experts cloned existing metric descriptions and iteratively changed all attributes to appropriately document new metrics. Nevertheless, this is not an intended behavior. Hence, the corresponding attribute definitions of the type metric description have to be adjusted to *optional*, except the title of a metric, which should remain a required attribute. In this manner, a new metric can be defined based on a title, whereas all other MMFS attributes can be modified after the initial creation of the metric description instance.

2. Further, the experts stated the tool could provide more support for the upload of the image, which represents the information model of a metric. Currently, users need to upload an image to our prototype, i.e., to attach it to the given metric description instance as a file and then to link the image by editing the information model attribute. According to the experts, it will be more convenient from a user's perspective, if this upload could be implemented as a *drag-and-drop* action.
3. The experts reported that the graphical representation of the attributes of the type metric description are too small in the *edit mode*. Thereby, our prototype uses the underlying Tricia functionality for editing pages. However, from user's perspective, the normal view of the metric description attributes, e.g. the calculation rule, is adequately sized to its content. Nevertheless, in edit mode, the attributes are represented only by a text field, which makes the editing process confusing. Hence, the experts recommended to use text areas instead of text fields for the edit mode of pages.

Concluding the talk, both experts reported high interest in applying our prototype in their daily business, if a corresponding commercial version of the software is made available. The experts further reported that the tool covers all their expectations for a holistic metric lifecycle management. Furthermore, one of the experts said: "During the application of your method, I realized that I would start now defining an EA management function in a completely different way, as I did before. By starting with the identification of the concrete demand from my management, I would first focus on understanding their goals and concerns and I will define corresponding metrics, allowing a quantitative view on the implementation progress of the initiative. Further, following your method, I get a minimal integrated information model resulting from the set of defined metrics, where each type, each attribute and each relationship of this EA model is linkable to concrete stakeholder demands, goals and concerns. Hence I can be sure, that the management and the collection processes for the required data are aligned with the expectations and needs of my stakeholders. What we have did back in the day was to start collecting all available information first and then linking this data to the needs of our stakeholders. Your minimality-based and demand-driven approach seems to me way more beneficial. And again, in my understanding, your tool provides a very elegant and effective support for the management of EA management metric where they belong - within my EA management tool."

6.2.4. Summary

Based on the collected feedback in the evaluation of our prototype, both industry experts confirmed the validity and the appropriateness of all design and architectural decisions made during the implementation of our conceptual solution. Thereby, the majority of the implemented features were tested in a real-word application of the prototype by the experts and these features proved to be helpful and useful. Additionally, no feature was reported as unnecessary or not appropriately implemented. Moreover, no feature, preventing a sufficient software support for our conceptual solution, was reported by the experts. Further, the conducted case study provided valuable insights in new metrics in the domain, as well as insights in concrete known metrics, which were selected for usage by the given organization and hence, indicating the benefit in terms of sharing documented knowledge by our solution. Based on the valuable feedback collected in the evaluation workshop, specific improvement potentials for a next iteration of the prototype's design have been identified and documented. Although the prototype will not be productively used by the evaluation partner, since a commercial version of the system is not yet available, the experts clearly confirmed interest in using a corresponding software solution in their daily work.

Based on these results, an existing organization, with an established EA management function, was able to successfully implement organization-specific metrics using our prototype. Thus, we can falsify our antithesis, defined in the beginning of this section and hence, we are able to verify our research hypothesis. Nevertheless, this case study provides only an initial verification of hypothesis. Hence, future research should conduct additional case studies to provide a wider empirical base for the verification of our hypothesis and to identify possible shortcomings of the solution and concrete improvement potentials.

6.2.5. Threats to Validity

With respect to the validity of the evaluation results presented in the previous sections, one can say that these subjective results are influenced by our personal relationship to the experts. In fact, the first expert participated in several other interviews and research projects at our chair. Nevertheless, he confirmed to provide only objective feedback from his perspective. In addition, based on the long and relevant professional experience and knowledge of the expert, we understand his judgment as valuable for the purpose of the evaluation of our prototype.

The second expert was introduced to us first for this case study. Hence, no personal relationship has been established in advance. In addition, since the expert used the software prototype independently, there was no personal contact to any researchers at our chair. Accounting for the relevant working experience of the expert in the BI and data analytics domains, we consider her feedback as in particular valuable regarding the discussion of alternative usage of BI-based systems for the purpose of metric management in the EA management domain.

As mentioned in Chapter 1, during this evaluation we could not evaluate all phases of a holistic metric life-cycle management. Although the experiment covered the phases from the definition, organization-specific instantiation, implementation, and prototypical evaluation of metrics, we could not evaluate the management of metrics over a longer period of time. Firstly, the available time frame for the experiment was too short to conduct a corresponding long-term study. Secondly, hence our prototype is not available as supported commercial software, the evaluation partner was not able to use the prototype as a productive software. Hence, our prototypical solution has to be adopted by a commercial version of Tricia, to allow the productive usage and long-term monitoring of our metric management method in its full range.

CHAPTER 7

Critical Reflection and Future Research

7.1. Conclusion

In this thesis, we presented an integrated software support for a holistic life-cycle management of metrics in the domain of EA management. For this purpose, in the conceptual part of the thesis, we introduced four concrete scenarios for the application of metrics in the domain (cf. Chapter 3). Further, we introduced a terminology base for specific types of EA management metrics. Additionally, we presented a list of 26 risks and related countermeasures, which have to be accounted by a metric life-cycle management method. We further introduced a generic metric management fact sheet, tailored to its application in our domain and enabling a comprehensive documentation of metrics with a minimal number of included description elements. This fact sheet additionally supports an organization-specific configuration and instantiation of metrics. Further, we presented an organized collection of EA management metrics observed in industry and uniformly described by the application of our fact sheet. This collection supports the timely retrieval of recommended metric best-practices for specific EA management goals and concerns. Consequently, we presented a holistic metric life-cycle management method, integrated within the BEAMS framework to allow the management of both—EA qualitative and EA quantitative models under consideration of best-practices and the specific context of user organizations.

Based on the created conceptual artifacts, in Chapter 5 we presented the main contribution of this thesis - *an integrated software support for a holistic metric life-cycle management method* within a concrete EA management tool. Therefore, we firstly defined concrete requirements for the implementation of our concepts, e.g. the design of a domain-specific model-based query language (MxL) enabling the implementation of the method base's metrics in the selected EA management tool. We further described the application of the software prototype in two parallel research projects, where we observed specific limitations of our solution on the one hand. On the other hand, by talking to and learning from the involved industry partners in these research projects, we collected valuable input for our research, e.g. the documentation of additional EA management metric best-practices and the identification of specific features for our software solution. Based on the results of this initial evaluation and following the main principle of design science research, we have redesigned our solution and developed an improved second version of our prototype. This improved prototype has been evaluated by a German bank, which has tested our software as an alternative to its own developed software system for the management of EA management metrics. As the results of the evaluation confirm, the prototype proved to provide a sufficient support for a holistic metric life-cycle management in the EA management domain.

7.2. Critical Reflection and Future Research

Although our prototype proved to successfully support a holistic life-cycle management of metrics in the EA management domain (cf. Chapter 6), we are aware of specific limitations and shortcoming of our solution, as already partially described by Reschenhofer [Re13], Schätzlein [Sc14], and Mahabaleshwar et.al [Ma14].

7.2.1. Authorization in MxL Functions

Designed as a multi-user application, Tricia provides an authorization mechanism allowing to specify which user or (group of users) can *read*, *write* (*write* access, implies *read* access), or *administrate* (management of access rules, implies *write* access) a given Tricia object, e.g. page or document. Thus, Tricia supports a controlled access to the qualitative EA model elements and their instances.

Problem / Limitation

Currently, our prototype does not provide a corresponding mechanism to manage access rights for MxL functions. Consequently, each Tricia user is allowed to view all existing MxL functions in the system. Furthermore, each user is allowed to call (invoke) any MxL function without permissions and hence, it is not possible to define MxL expressions, which can be called only by specific user (or specific group of users).

Nevertheless, MxL expressions access the underlying qualitative EA model elements using Tricia's default authorization model for information objects. Therefore, the authorization rules defined on information objects are also applied on the access by MxL functions. Consequently, if a given user, calling a specific MxL expression, is not permitted to read the corresponding information model elements (e.g. specific type instances), these objects are filtered during the evaluation of the expression. For instance, if a given user is not permitted to read any instance of the type *Business application* and the user invokes the MxL function implementing metric *Application continuity plan availability*, the result will be *0*.

Proposed Solution

One possible solution for this shortcoming is to apply the existing authorization model for information objects on MxL functions. Thereby, the three permission levels described above require a slightly different behavior for MxL functions as follows:

Read access - allows the invocation of a MxL function, as well as the access to its definition, however, changes to the function's attributes, e.g. name, description, parameters, or method stub are not allowed.

Write access - allows the invocation as well as the management of MxL function's attributes, e.g. parameters and method stub. Currently, each user of the prototype has implicitly write access for each MxL function.

Administrate access - allows the management of permissions levels for MxL functions.

Further, the prototype requires an additional rule, concerned with authorization of users regarding the definition of new MxL functions. This rule however must be applied on an appropriate Tricia container object (e.g. a workspace) and not on a MxL provider (functions, derived attribute, and pages, containing embedded MxL expressions in their rich-text content). For example, a workspace may implement a role named *MxL creators* referring to all users allowed to define new MxL custom functions in this specific workspace.

7.2.2. Identity Evaluation in MxL

As described above, MxL uses the authorization mechanism of Tricia for accessing the underlying information objects. Hence, each MxL provider is invoked under the consideration of the executor's identity.

Problem / Limitation

The current system behavior implies that the evaluation of a given MxL expression returns different results for users with different access rights (cf. Section 7.2.1). While this system behavior can be considered as useful for specific scenarios (e.g. the definition of user-specific views), the definition and evaluation of metrics requires a different approach, since metrics have to return the same value for all users (i.e., the metric owner, responsible person and data contact) involved in the management process of a given metric instance.

Proposed Solution

To address this problem, the following identity evaluation strategies should be taken into account by future research:

Executor's identity This is the current identity evaluation approach in Tricia. Thereby, the function executor's identity is used for the evaluation of the MxL expression.

Definer's identity A more prevalent approach for the purpose of a holistic metric life-cycle management is the execution of an expression by considering the definer's (author's) identity.

Specified user identity In this strategy, the creator of a MxL provider can explicitly specify the identity of a concrete user, whose identity is used for the evaluation.

System identity In this strategy, a specific system identity is defined, which is used for the evaluation of MxL expressions. For example, this identity might be the web application's process identity.

In our understanding, to resolve the current shortcoming, a mix of the currently implemented executor's identity strategy and one from the other three alternatives, e.g. definer's identity, may prove sufficient to support both—user-specific and user-independent definition of MxL providers (i.e., custom functions, derived attributes and embedded expression in rich-text content).

7.2.3. Evaluation Strategy of MxL Functions

As described in Chapter 5, MxL expressions are currently evaluated at runtime. This however leads to long response/computation times in specific cases.

Problem / Limitation

While the current approach is appropriate for trivial functions based on a simple information model with a small number of instance data, recursive functions, which are computed on complex information models with a large amount of instance data, lead to (unacceptably) long computation/response times. According to our experience by the application of the prototype in the *CALM*³ project, we consider this limitation as problematic from a user's perspective.

Proposed Solution

In our understanding, the following four MxL evaluation performance improvement strategies should be taken into account by future research to improve the observed problem:

On demand In this strategy, a MxL provider is evaluated at runtime whenever the function is been invoked. This is the currently implemented evaluation strategy in Tricia.

Cached In this strategy, a MxL provider is evaluated at runtime only for the first invocation of the provider. Thereafter, the computed value is stored within an in-memory cash, allowing to pass the cashed value to all following requests. Nevertheless, since currently the executor's identity evaluation strategy is implemented in Tricia, expression values must be cashed with respect to the executor's identity to ensure proper results. Further, whenever a relevant change is performed (either in the provider's code, or in the underlying information model and its instances), the cashed values need to be recomputed.

Scheduled In contrast to caching, this evaluation strategy allows a regular evaluation of MxL expression using a configurable schedule. In addition, by implementing this strategy, Tricia could persist the computed values and hence, the system can allow the historicization of MxL expression values.

On change In contrast to *on demand*, this strategy triggers the evaluation of MxL expressions whenever a change in the underlying information model occurs. This strategy combined with the cashing strategy to ensure the validity of cashed values with respect to changes in the data model might prove to be a sufficient solution for the given problem.

Problem / Limitation

As discussed above, Tricia currently does not persist calculated MxL expression values. Hence, users need to employ the @ operator to obtain metric historicization values. Nevertheless, this leads to additional evaluations of the given expression on corresponding past information model states, which consequently leads to even longer response/computation times.

Proposed Solution

To address this problem, future research should investigate the idea of storing (persisting) the MxL expressions history in the system. Thereby, if for instance the @ operator is called for a given MxL function, the system can check weather appropriate historicization data is already persisted and can be retrieved from the underlying data base.

Problem / Limitation

As described in Chapter 5, Tricia supports the definition and computation of metric visualizations at run-time. Nevertheless, the underlying MxL expressions, which in addition might use the @ operator, are currently evaluated at run-time.

Proposed Solution

To address this problem, future research should explicitly account for the usage of metric visualizations during the performance optimizations described in the previous paragraphs. Currently, whenever a metric visualization is called, its response time is determined by the sum of both—the computation time of the underlying query and the rendering time for the graphical representation. Additionally, since managers are prone to define dashboards consisting of several visualizations, this issue can lead to longer and even unacceptable response times. Therefore, it seem reasonable to allow visualizations definers to specify a schedule for both—the calculation and the historicization of the underlying MxL expressions. Further, such schedulers can be implemented as night batch jobs, shifting long and complex computations during the times when the users are not working with the system. Additionally, every visualization can be understood as a known information demand of a specific user (or group). Hence, it seems reasonable to cash the current values of the underlying MxL functions, too. By the combination of these two strategies, the underlying computation times can be significantly reduced, since the metric historicization values can be loaded by the data base and the current metric value can be retrieved from in-memory cash.

7.2.4. Version Control of MxL Functions

Tricia provides a version control mechanism for its objects. Hence, all changes performed by users on a given Tricia object are persisted. Further, Tricia supports a (text-based) comparison of different object versions empowering users to understand performed changes, and (if write access is granted), to revert the object to a specific previous version.

Problem / Limitation

In our prototype, MxL providers (e.g. custom functions) are not covered by a corresponding version control. Even more, since currently every user has (implicit) write access to all MxL functions, the evolution of expressions cannot be managed in a transparent manner, which we consider as a critical limitation of the prototype. In the context of metrics management, a responsible person for a metric could currently easily manipulate the calculation rule of a given metric (for instance by overwriting the method stub to return a static number), and ensure that every user will get the same result (independent from the used identity). Thereby, in the current implementation of the prototype, the system is not able to trace this change back to the user.

Proposed Solution

To address this problem, Tricia should ensure transparency of the changes performed on MxL providers. Therefore, the existing mechanism for version control of information objects could be applied on all MxL providers. Further, Tricia supports a so-called feature *watch*, empowering users to configure and to use a notification service (via e-mail) for reporting changes of specific Tricia information objects. Hence, this notification service could be extended to support a watch functionality for MxL providers, too. In this manner, all interested users in a given MxL function can be automatically notified whenever corresponding changes are performed.

7.2.5. Import of EA Model History Data

Tricia allows by its design the import of information objects originating from extern sources, e.g. by parsing an Excel file, where both—the information model's schema and its instance data are described. Thereby, Tricia creates a corresponding hybrid representation of the model and its instance data.

Problem / Limitation

According to Ahlemann et al. [Ah12], the consideration of EA model historicization data is indispensable for the management and analysis of the EA's evolution. Nevertheless, Tricia currently does not support the import of historicization data. From a technical point of view, the import can be invoked multiple times. Thereby, if a hybrid representation for the given schema and its instance data already exist, the imported data is stored in a new version of the corresponding Tricia objects. However, the time-stamps of this model history changes are managed by Tricia, which documents its system's time, when the import was performed. Hence, it is not possible to import the history of a given EA model for specific points in time by performing multiple Excel imports in terms of model history snapshots.

Proposed Solution

In our understanding, future research should investigate different strategies for both—import of historicization data from related research fields (e.g. IT controlling and BI) and the integration of external data sources by logical data integration using corresponding techniques, e.g. the *open data protocol* (OData). Nevertheless, both approaches have to deal with different challenges. For instance, by implementing an import mechanism for EA model historicization data import, Tricia cannot link the corresponding hybrid model changes to concrete users. Hence, the system can only indicate that the data was imported. In the case of logical data integration, Tricia users seem prone to enrich the integrated model data, e.g. with additional hybrid attributes. Nevertheless, if the integrated model changes in the master source (e.g. a specific schema element is deleted), Tricia has to account for this change (e.g. by deleting the corresponding Tricia schema model element, its instances and corresponding relationships to other object). This however might not be the intended system behavior by Tricia's users. Additionally, users might perform changes to the integrated model in Tricia and might request the propagation of their changes to the underlying master source, which however leads to several new data integration issues. Currently, both alternatives are the subject of research in two ongoing research projects at our chair.

7.2.6. Integration of the MMFS Structure as a Tricia Object

Currently, our MMFS structure is implemented in the prototype as a type (*metric description*), and each metric of the catalog is documented as an instance of this type (cf. Section 5.1.4). Further, to implement a metric's calculation rule, we defined the additional attribute *formula*, which links an implementing MxL custom function with a metric description instance. Further, the majority of the MMFS elements (e.g. target value) are represented by text elements.

Problem / Limitation

Although our current solution proved to be appropriate for the support of a holistic life-cycle management of organization-specific metrics (cf. Chapter 6), the current implementation does not incorporate all possible benefits by the performed Tricia integration.

Solution

According to our understanding, the MMFS structure can be alternatively implemented as a new Tricia object. Thereby, the following benefits can be achieved:

1. Tricia's latest version (3.9) provides a feature allowing the UML representation of the hybrid model of a given workspace as a scalable vector graphic at runtime. By integrating

the MMFS structure as a Tricia object, this functionality can be used and hence, the process of creating and uploading an information model for a MMFS instance can be fully automated. Thereby, the information model can be computed based on the metric description's formula.

2. Since the MMFS structure requires the documentation of an owner, responsible person, and data contacts for a metric, these elements can be linked to concrete Tricia users. With respect to the previously described MxL provider access management limitation, the system can ensure read access to the related MxL function only for involved stakeholders and further - the system can grant write access only to the metric owner. Additionally, Tricia can ensure automated notification of the readers, whenever the owner performs a change to a metric description. Furthermore, Tricia can use the owner's identity for the evaluation of MMFS objects to ensure, that all readers see the same metric evaluation results.

3. The integration of the MMFS structure as Tricia object can further account for the visualization of metrics. For instance, one additional attribute can be introduced to define a common visualization of a given metric for all involved users. Further, the documented target and planned metric values can be automatically considered by the visualization to indicate the status towards the achievement of these values. Moreover, the interpretation rule of the MMFS can be automatically considered by the visualization, e.g. by a bullet chart indicating qualitative interpretation of a measured metric as *good*, *acceptable*, or *problematic*.

4. According to our solution, organizations start with the application of our metric management method by cloning the catalog's workspace. Thereby, the first column of the mapping table of each MMFS instance can be automatically filled with the recommended model element names according to the catalog. Thereafter, whenever a change in the terminology is made by users (in the underlying calculation rule, e.g. the term *business application* is renamed to *application*), the mapping table can be automatically updated. If the underlying information model changes as a consequence of a change performed by the metric's owner in the calculation rule, the mapping table can be automatically updated ensuring the consistency between the information model and the mapping table.

5. Based on the MMFS element *frequency of measurement*, the system can automatically evaluate MMFS instances to support the reporting process of the metric results. Moreover, a corresponding notification service can be used to inform all involved stakeholders regarding the measured values.

6. As described in Chapter 3, the prototype can support a learning method for our method base (metrics best-practices). Thereby, cloned workspaces can be regularly examined to obtain usage data for the catalogs metrics. Thus, this analysis can concentrate on the identification of selected, adapted, deleted, and new defined metric by the user organizations of the catalog. Furthermore, the users can actively give feedback for specific metrics, e.g. reporting an anti-pattern candidate, and ideally, the users can share metric usage data to allow the quality assessment of metrics as described in Section 4.2.

Hence, future research should investigate the question, if the implementation of the MMFS structure as a new Tricia object is more helpful and useful than the current implementation of the structure as a type.

7.2.7. Investigating MxL from Computability Theory Perspective

As described in Chapter 5, MxL was designed to provide a minimal and sufficient number of language constructs, required for the implementation of the catalog's metrics.

Problem / Limitation

During our research, the question of examining MxL from a computability theory point of view was not in the scope our work. Hence, it seems reasonable to study the computability capabilities of MxL and to examine the need of redesigning the language to support additional computation capabilities.

Solution

Several widely used object-oriented languages, e.g. Java and C++, as well as several functional programming languages, e.g. F# and Haskell, are turing-complete. Since MxL combines language paradigms from both types of programming languages, it will be interesting to answer the question, if MxL is turing-complete. In addition, accounting for its design purpose, it sounds reasonable to investigate the question by future research, if turing-completeness is a relevant language design requirement for MxL or not.

APPENDIX A

Questions List for the Evaluation

The subsequent Table A.1 shows all 50 predefined questions which were asked during the interview in the evaluation workshop. Thereby, table rows colored in gray indicate context information, which have been provided by the interviewer, whereas rows colored in white indicate the questions, which have been asked.

Question
MxL 2.0 design evaluation
MxL is designed to provide a sufficient and minimal expressiveness for the calculation of the catalogs metrics.
What is your opinion regarding the design decision to differentiate between basic and user-defined custom functions in MxL?
What are recommendations, remarks and comments regarding the expressiveness (basic types, functions and operators) of MxL?
MxL is designed as a functional language.
What is your opinion regarding the decision to design MxL as a functional language (to support higher-order functions and (implicit) lambda expressions)?
How difficult is it to learn / use these functional features?
Have you used these functional features for the implementation of your metrics? What is your experience?
MxL is designed to support access to the underlying EA model history.
What is your opinion regarding the design decision to allow model history access in MxL functions?
Have you used / do you plan to use this feature for the management of your organization-specific metrics? What is your experience?
MxL is designed to be (static) type-safe.
What is your opinion regarding the design decision to ensure type-safety in MxL?

Question
What is your opinion regarding the design decision to perform type-checking of MxL functions at compile time?
MxL is designed to be object-oriented.
What is your opinion regarding the design decision to use object-oriented language paradigms in MxL?
How difficult is it to understand and to apply these programming language paradigms?
MxL is designed to support recursive functions (transitive closure).
What is your opinion regarding the design decision to support recursive functions in MxL?
Have you used recursive functions for the implementation of your organization-specific metrics? What is your experience?
MxL is designed to support both – forward and backward navigation.
What is your opinion regarding the design decision to support both - forward and backward navigation in MxL?
Have you used both - forward and backward navigation for the implementation of your organization-specific metrics? What is your experience?
MxL was designed to fulfill the predefined requirements as already discussed.
What are missing language design requirements?
What is your opinion regarding the design decision to apply a model-based query language for the computation of EA management metrics?
Do you have any recommendations for the improvement of the overall language design?
How helpful was the documentation of MxL and the provided code examples for the implementation of your organization specific metrics?
Evaluation of TxL 2.0 and the related Tricia extensions
What is your opinion regarding the design decision to integrate MxL in an EA management tool?
What are benefits / disadvantages regarding the idea to manage metrics in an EA management tool compared to Excel, BI tools, or self-developed solutions?
User-defined metric visualizations
What is your opinion regarding the design decision to support user-defined metric visualizations in Tricia?
What is your opinion regarding the design decision to support the embedding of visualizations in the rich-text content of Tricia pages?
Metric relationships management.
What is your opinion regarding the design decision to show metric relationships in the view of a given MxL custom function / derived attribute?
Do you miss any further types of relationships between MxL elements?
Do you miss any further user decision support options when possible inconsistencies (on the deletion of MxL function) are detected by the system, beside the options *cancel*, *cascade deletion of referencing MxL providers* and *ignore possible inconsistencies*?
What user decision support options do you require, when a possible inconsistence (based on performed MxL code changes is detected by the system) (Currently, the inconsistencies are only marked with a corresponding symbol)?

Question
Web-based MxL development support.
What is your opinion regarding the design decision to provide a web-based support for the development of MxL functions?
What is your opinion regarding the usability of the MxL query editor?
What is your opinion regarding the usability of the expressions completion support?
What is your opinion regarding the offered syntax highlighting?
What are improvement recommendations for the provided syntax highlighting?
What is your opinion regarding the provided "Test expression" functionality?
What is your opinion regarding the shown MxL error messages by the system?
Evaluation of the provided software support for our holistic metric life-cycle management method
What is your opinion regarding the design decision to implement the MMFS structure as a type ("metric description")?
What is your opinion regarding the design decision to link metric descriptions to corresponding custom MxL functions by the additional attribute "formula" in the type metric description?
What is your opinion regarding the design decision to allow the customization of Tricia's types at runtime?
Have you used / Do you plan to use this functionality for your organization? What is your experience?
What is your opinion regarding the design decision to implement the metric catalog as a Wiki using the type metric description?
What is your opinion regarding the provided navigation support for the metric catalog implementation (Goal-Concern-Metric matrix)?
What is your opinion regarding the design decision to initialize the usage of our holistic management method with a clone of the metric catalog implementation?
Which metric from the catalog have you used for your metric system (without adaptations)?
Which metrics from the catalog have you adapted?
Which metrics from the catalog have you deleted?
Which new metrics have you defined?
Which difficulties have you faced during the documentation of metrics using the *metric description* type?
What steps, artifacts and roles from our conceptual metric management method are not supported by the prototype?
Would you use a commercial version of this prototype for your daily work?
Compared to your current software solution, what are the major advantages and disadvantages of our prototype?
What features do you miss in the prototype?

Table A.1.: Complete list of questions asked in the thesis' evaluation

Bibliography

[Ad10] Addicks, J. S.: *Bewertung betrieblicher Anwendungen im Kontext ihrer Unternehmensarchitektur.* PhD thesis. Carl von Ossietzky University of Oldenburg. 2010.

[AGW11] Aier, S.; Gleichauf, B.; Winter, R.: *Understanding Enterprise Architecture Management Design-An Empirical Analysis.* In *Wirtschaftsinformatik.* page 50. 2011.

[Ah06] Aho, A. V.; Lam, M. S.; Sethi, R.; Ullman, J. D.: *Compilers. Principles, Techniques and Tools.* Addison Wesley. 2006.

[Ah12] Ahlemann, F.; Stettiner, E.; Messerschmidt, M.; Legner, C.: *Strategic Enterprise Architecture Management.* Springer-Verlag. 2012.

[ARW08a] Aier, S.; Riege, C.; Winter, R.: *Classification of Enterprise Architecture Scenarios – An Exploratory Analysis. Enterprise Modelling and Information Systems Architectures.* 3:14–23. 2008.

[ARW08b] Aier, S.; Riege, C.; Winter, R.: *Unternehmensarchitektur – Literaturüberblick Stand der Praxis. Wirtschaftsinformatik.* 50(4):292–304. 2008.

[BC07] Berg-Cross, G.: *Improving representation and conceptualization for enterprise architectures.* In *International Conference on Enterprise System Theory.* 2007.

[BCR94] Basili, V. R.; Caldiera, G.; Rombach, H. D.: *The Goal Question Metric Approach.* Wiley. New York. 1994.

[BH07] Box, D.; Heijlsberg, A.: *LINQ: .NET Language-Integrated Query.* `http://msdn.microsoft.com/en-us/library/bb308959.aspx`. 2007.

[Bi05] Bird, S. M.; David, C.; Farewell, V. T.; Harvey, G.; Tim, H.; Peter, C.; others: *Performance indicators: good, bad, and ugly. Journal of the Royal Statistical Society: Series A (Statistics in Society).* 168(1):1–27. 2005.

[BJS13] Buschle, M.; Johnson, P.; Shahzad, K.: *The Enterprise Architecture Analysis Tool–Support for the Predictive, Probabilistic Architecture Modeling Framework.*

In *Proceedings of the 19th Americas Conference on Information Systems (AMCIS) :*. 2013. QC 20131101.

[BMN10] Büchner, T.; Matthes, F.; Neubert, C.: *Data Model Driven Implementation of Web Cooperation Systems with Tricia. 3rd International Conference on Objects and Databases.* 2010.

[Bo03] Bourne, M.; Neely, A.; Mills, J.; Platts, K.: *Implementing performance measurement systems: a literature review. International Journal of Business Performance Management.* 5(1):1–24. 2003.

[BP00] Black, J.; Porter, L.: *Management: Meeting New Challenges.* Prentice-Hall International. 2000. 9780321014078.

[BS11] Buckl, S.; Schweda, C. M.: *On the State-of-the-Art in Enterprise Architecture Management Literature.* Technical report. Technische Universität München. München, Germany. 2011.

[Bu98] Butler, T.: *Towards a hermeneutic method for interpretive research in information systems. Journal of Information Technology.* 13:285–300. 1998.

[Bu06] Burger, T.: *Management Cockpit: Unterstützung von Kennzahlen durch eine effiziente Visualisierung.* LFI, Abt. für Handwerkswissenschaften. 2006.

[Bu08] Buckl, S.; Ernst, A. M.; Lankes, J.; Matthes, F.: *Enterprise Architecture Management Pattern Catalog (Version 1.0, February 2008).* Technical report. Chair for Informatics 19 (sebis), Technische Universität München. Munich, Germany. 2008.

[Bu09] Buckl, S.; Ernst, A. M.; Lankes, J.; Matthes, F.; Schweda, C. M.: *State of the Art in Enterprise Architecture Management 2009.* Technical report. Chair for Informatics 19 (sebis), Technische Universität München. Munich, Germany. 2009.

[Bu10] Buckl, S.; Dierl, T.; Matthes, F.; Schweda, C. M.: *Building Blocks for Enterprise Architecture Management Solutions.* In (Harmsen, F. e. a., Ed.): *Practice-Driven Research on Enterprise Transformation, second working conference, PRET 2010, Delft.* pages 17–46. Berlin, Heidelberg, Germany. 2010. Springer.

[Bu11] Buckl, S. M.: *Developing Organization-Specific Enterprise Architecture Management Functions Using a Method Base.* PhD thesis. Technische Universität München. München, Germany. 2011.

[Cl85] Cleveland, W. S.: *The elements of graphing data.* Wadsworth advanced books and software Monterey, CA. 1985.

[De06] Demenchuk, A.: *Beaver-a lalr parser generator.* `http://beaver.sourceforge.net/`. 2006.

[Do00] von Dobschütz, L.: *IV-Controlling.: Konzepte, Umsetzungen, Erfahrungen.* Wissenschaft & Praxis. Gabler, Betriebswirt.-Vlg. 2000. 9783409116770.

[Dr06] Drucker, P. F.: *The Practice of Management.* Harper Paperbacks. Oxford, UK. Reissue edition. 2006.

[Dr10] Drucker, P.: *The Practice of Management.* HarperCollins. 2010. 9780062005441.

[Ec90] Eccles, R. G.: *The performance measurement manifesto. Harvard business review.* 69(1):131–137. 1990.

[Ec10] Eckerson, W. W.: *Performance dashboards: measuring, monitoring, and managing your business.* John Wiley & Sons. 2010.

[Er10] Ernst, A. M.: *A Pattern-Based Approach to Enterprise Architecture Management.* PhD thesis. Technische Universität München. München, Germany. 2010.

[Fe05] Few, S.: *Keep Radar Graphs Below the Radar - Far Below.* http://www.perceptualedge.com/articles/dmreview/radar_graphs.pdf. 2005.

[Fe06] Few, S.: *Information dashboard design.* O'Reilly. 2006.

[Fe07] Few, S.: *Save the pies for dessert. Perceptual Edge Visual Business Intelligence Newsletter.* pages 1–14. 2007.

[Fe09a] Feldschmid, A.: *Konzeption und prototypische Implementierung eines Steuerungscockpits im Kontext des Managements von Unternehmensarchitektur.* Master's thesis. Hochschule Rosenheim. 2009.

[Fe09b] Few, S.: *Now you see it: simple visualization techniques for quantitative analysis.* Analytics Press. 2009.

[FGM07] Franceschini, F.; Galetto, M.; Maisano, D.: *Management by Measurement: Designing Key Indicators and Performance Measurement Systems.* Springer. 2007. 9783540732129.

[FL00] Frazer, L.; Lawley, M.: *Questionnaire design & administration: A practical guide.* Wiley New York. 2000.

[Fo68] Forrester, J. W.: *Principles of Systems.* Wright-Allen Press. 1968.

[Fo71] Forrester, J. W.: *World dynamics.* Wright-Allen Press. 1971.

[Fo90] Forrester, J.: *Principles of systems.* System dynamics series. Productivity Press. 1990. 9780915299874.

[Fo94] Forrester, J. W.: *System dynamics, systems thinking, and soft OR. System Dynamics Review.* 10(2-3):245–256. 1994.

[GL06] Gläser, J.; Laudel, G.: *Experteninterviews und qualitative Inhaltsanalyse.* VS Verlag. 2006. 9783531150666.

[Go13] Gosling, J.; Joy, B.; Steele, G.; Bracha, G.; Buckley, A.: *The Java Language Specification,: Java Se 7 Ed.* Always learning. Prentice Hall PTR. 2013. 9780133260229.

[Gr73] Graham, A. K.: *Principles on the relationship between structure and behavior of dynamic systems.* Dissertation. Massachusetts Institute of Technology. Cambridge. 1973.

[Gr09] Gringel, P.: *Metriken zur Bewertung von Anwendungslandschaften.* Master's thesis. Universität Oldenburg. 2009.

[Gr12] Group, T. O.: *TOGAF Version 9.1: Section 2.2.* http://pubs.opengroup.org/architecture/togaf9-doc/arch/chap02.html. October 2012.

[Ha13] Hauder, M.; Roth, S.; Matthes, F.; Lau, A.; Matheis, H.: *Supporting collaborative product development through automated interpretation of artifacts.* In *Proceedings of the BMSD 2013 – Third International Symposium on Business Modeling and Software Design.* 2013.

[He96] Heidegger, M.: *Being and time: A translation of Sein und Zeit.* SUNY Press. 1996.

[He04] Hevner, A. R.; March, S. T.; Park, J.; Ram, S.: *Design Science in Information Systems Research. MIS Quarterly.* 28(1):75–105. 2004.

[HK98] Hauser, J.; Katz, G.: *Metrics: you are what you measure! European Management Journal.* 16(5):517–528. 1998.

[In07] International Organization for Standardization: *ISO/IEC 42010:2007 Systems and software engineering – Recommended practice for architectural description of software-intensive systems.* 2007.

[IT09] ITGI: *CobiT 4.1.* Technical report. IT Governance Institute. Rolling Meadows, IL, USA. 2009.

[JG02] Jäger-Goy, H.: *Führungsinstrumente für das IV-Management.* Lang. Frankfurt am Main. 2002. 3-631-37811-4.

[Jo13a] Johnson, P.; Ullberg, J.; Buschle, M.; Franke, U.; Shahzad, K.: *P2AMF: Predictive, Probabilistic Architecture Modeling Framework. Enterprise Interoperability.* pages 104–117. 2013.

[Jo13b] Josey, A.: *ArchiMate 2.1.* The Open group series. Haren Publishing, Van. 2013. 9789401805032.

[KA10] Krause, H.; Arora, D.: *Controlling-Kennzahlen - Key Performance Indicators: Zweisprachiges Handbuch Deutsch/Englisch - Bi-lingual Compendium German/English.* Oldenbourg Wissenschaftsverlag. 2010. 9783486596908.

[KAV05] Kaisler, S.; Armour, F.; Valivullah, M.: *Enterprise Architecting: Critical Problems.* In *Proceedings of the Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05)-Track 8-Volume 08.* Washington, DC, USA. 2005. IEEE Computer Society Washington, DC, USA.

[Ke07] Keller, W.: *IT-Unternehmensarchitektur.* dpunkt.verlag. Heidelberg, Germany. 2007.

[Ke13] Kerzner, H. R.: *Project management metrics, KPIs, and dashboards: a guide to measuring and monitoring project performance.* John Wiley & Sons. 2013.

[Kl10] Klein, G.: *Jflex user's manual. Available on-line at*`http://jflex.de/`. 2010.

[KN91] Kaplan, R. S.; Norton, D. P.: *The Balanced Scorecard – Measures That Drive Performance. Harvard Business Review.* 70(1):71–79. 1991.

[Ko95] Kotter, J. P.: *Leading change: Why transformation efforts fail. Harvard business review.* 73(2):59–67. 1995.

[Kü10] Kütz, M.: *Kennzahlen in der IT. Werkzeuge für Controlling und Management.* dpunkt.verlag. Heidelberg, Germany. 4th edition. 2010. 978-3898647038.

[Kü13] Kütz, M.: *IT-Controlling für die Praxis - Konzepte und Methoden*. dpunkt.verlag. 2013. 9783864900037.

[La08] Lankes, J.: *Metrics for Application Landscapes – Status Quo, Development, and a Case Study*. PhD thesis. Technische Universität München, Fakultät für Informatik. Munich, Germany. 2008.

[La13] Lagerström, R.; Baldwin, C. Y.; Maccormack, A. D.; Aier, S.: *Visualizing and Measuring Enterprise Application Architecture: An Exploratory Telecom Case*. Technical report. Harvard Business School. 2013.

[LDH08] Lawson, R.; Desroches, D.; Hatch, T.: *Scorecard best practices: design, implementation, and evaluation*. John Wiley & Sons. 2008.

[LE02] Lebas, M.; Euske, K.: *A conceptual and operational delineation of performance. Business performance measurement: Theory and practice*. pages 65–79. 2002.

[LE06] Levy, Y.; Ellis, T. J.: *A Systems Approach to Conduct an Effective Literature Review in Support of Information Systems Research. Informing Science*. 9. 2006.

[LKL10] Lucke, C.; Krell, S.; Lechner, U.: *Critical Issues in Enterprise Architecting - A Literature Review*. In *Proceedings of the Sixteenth Americas Conference on Information Systems (AMCIS 2010)*. Lima, Peru. 2010.

[LLO93] Luftman, J. N.; Lewis, P. R.; Oldach, S. H.: *Transforming the enterprise: The alignment of business and information technology strategies. IBM Systems Journal*. 32(1):198–221. 1993.

[Ma08] Matthes, F.; Buckl, S.; Leitel, J.; Schweda, C. M.: *Enterprise Architecture Management Tool Survey 2008*. Chair for Informatics 19 (sebis), Technische Universität München. Munich, Germany. 2008.

[Ma12a] Matthes, F.; Monahov, I.; Schneider, A.; Schulz, C.: *EAM KPI Catalog v1.0*. `http://wwwmatthes.in.tum.de/wikis/sebis/eam-kpi-catalog(cited2012-06-15)`. 2012.

[Ma12b] Matthes, F.; Monahov, I.; Schneider, A.; Schulz, C.: *A Template-Based Design Method to Define Organization-Specific KPIs for the Domain of Enterprise Architecture Management*. In (Büren, G.; Dumke, R. R.; Ebert, C.; Münch, J., Ed.): *Metrikon 2012 – Praxis der Software-Messung*. pages 203–214. Shaker Verlag, Aaachen. 2012.

[Ma12c] Matthes, F.; Monahov, I.; Schneider, A.; Schulz, C.: *Towards a unified and configurable structure for EA management KPIs*. 2012.

[Ma13] Matheis, H.: *SmartNet Navigator and application guidelines. Sehenth Framework Programme*. 2013. SmartNets - The Transformation from Collaborative Knowledge Exploration Networks into Cross Sectoral and Service Oriented Integrated Value Systems.

[Ma14] Mahabaleshwar, M.: *FAnalyzing the evolution of EA metrics – scenarios and requirements for an integrated tool support*. Guided research. Fakultät für Informatik, Technische Universität München. 2014.

[Me08] Meadows, D. H.: *Thinking in systems: A primer.* Chelsea Green Publishing. 2008.

[MLA10] McAffer, J.; Lemieux, J.; Aniszczyk, C.: *Eclipse Rich Client Platform.* Eclipse Series. Pearson Education. 2010. 9780321612342.

[MN11] Matthes, F.; Neubert, C.: *Using Hybrid Wikis for Enterprise Architecture Management.* In *7th International Symposium on Wikis and Open Collaboration (WikiSym).* Mountain View, California, USA. 2011.

[MNS11] Matthes, F.; Neubert, C.; Steinhoff, A.: *Hybrid Wikis: Empowering Users to Collaboratively Structure Information.* In *ICSOFT (1).* pages 250–259. 2011.

[Mo11] Monahov, I.: *Supporting enterprise performance management with organization-specific indicators.* In (Norta, A.; Ruohomaa, S., Ed.): *EDOC2011 PhD Student Symposium Proceedings.* volume B-2011-1 of *Department of Computer Science Series of Publications B.* pages 12–16. University of Helsinki, Department of Computer Science. 2011.

[MRM13] Monahov, I.; Reschenhofer, T.; Matthes, F.: *Design and prototypical implementation of a language empowering business users to define Key Performance Indicators for Enterprise Architecture Management.* In *Enterprise Distributed Object Computing Conference Workshops (EDOCW), 2013 17th IEEE International.* pages 337–346. IEEE. 2013.

[MSM13] Monahov, I.; Schumaiev, K.; Matthes, F.: *Sebis Study: Cloud Adoption and Strategy 2013, version 0.9.* 2013.

[NAK02] Neely, A.; Adams, C.; Kennerley, M.: *The performance prism: the scorecard for measuring and managing business success.* Financial Times Series. Financial Times/Prentice Hall. 2002. 9780273653349.

[Ne99] Neely, A. D.: *The performance measurement revolution: why now and what next? International Journal of Operations & Production Management.* 19(2):205–228. 1999.

[Ne02] Neely, A. D.: *Business performance measurement: theory and practice.* Cambridge University Press. 2002.

[Ne12] Neubert, C.: *Facilitating Emergent and Adaptive Information Structures in Enterprise 2.0 Platforms.* Dissertation. Technische Universität München. München. 2012.

[NGP95] Neely, A. D.; Gregory, M.; Platts, K.: *Performance measurement system design: a literature review and research agenda. International journal of operations & production management.* 15(4):80–116. 1995.

[Ni08] Niemi, E.: *Enterprise architecture benefits: Perceptions from literature and practice. Evaluation of enterprise and software architectures: critical issues, metrics and practices:[AISA Project 2005-2008]/Eetu Niemi, Tanja Ylimäki & Niina Hämäläinen (eds.). Jyväskylä: University of Jyväskylä, Information Technology Research Institute, 2008.-(Tietotekniikan tutkimusinstituutin julkaisuja, ISSN 1236-1615; 18). ISBN 978-951-39-3108-7 (CD-ROM).* 2008.

[Ob11a] Object Management Group (OMG): *Business Process Model and Notation (BPMN), Version 2.0 (formal/2011-01-03)*. http://www.omg.org/spec/BPMN/2.0/. 2011.

[Ob11b] Object Management Group (OMG): *OMG Unified Modeling Language^{TM}(OMG UML), Superstructure – Version 2.4.1 (formal/2011-08-05)*. http://www.omg.org/spec/UML/2.4.1/. 2011.

[Ob12] Object Management Group (OMG): *OMG Object Constraint Language (OCL)*. http://www.omg.org/spec/OCL/2.3.1/. 2012.

[Of00] Office of Government Commerce (OGC): *ITIL – Service Delivery*. IT Infrastructure Library (ITIL). The Stationery Office. Norwich, UK. 2000.

[OP10] Osterwalder, A.; Pigneur, Y.: *Business model generation: a handbook for visionaries, game changers, and challengers*. John Wiley & Sons. 2010.

[Pa07] Parmenter, D.: *Key Performance Indicators: Developing, Implementing,and Using Winning KPIs*. Wiley. 2007. 0-470-095881.

[Pa10] Parmenter, D.: *Key performance indicators (KPI): developing, implementing, and using winning KPIs*. John Wiley & Sons. 2010.

[Pe98] Perrin, B.: *Effective use and misuse of performance measurement. American journal of Evaluation*. 19(3):367–379. 1998.

[Pr12] Probst, H.: *Kennzahlen: Richtig anwenden und interpretieren*. Alles, was Sie wissen müssen. Redline Wirtschaft. 2012. 9783868813425.

[PS10] Popova, V.; Sharpanskykh, A.: *Modeling organizational performance indicators. Information Systems*. 35(4):505–527. 2010.

[PSP12] Plessius, H.; Slot, R.; Pruijt, L.: *On the Categorization and Measurability of Enterprise Architecture Benefits with the Enterprise Architecture Value Framework*. In *Trends in Enterprise Architecture Research and Practice-Driven Research on Enterprise Transformation*. pages 79–92. Springer. 2012.

[Ra13] Rassa, J.: *Nutzungsmöglichkeiten von Modellen der Unternehmensarchitektur bei der Entwicklung einer Balanced Scorecard für eine IT Organisation*. Master's thesis. Fakultät für Informatik, Technische Universität München. 2013.

[Re13] Reschenhofer, T.: *Design and prototypical implementation of a model-based structure for the definition and calculation of Enterprise Architecture Key Performance Indicators*. Master's thesis. Fakultät für Informatik, Technische Universität München. 2013.

[RMM14] Reschenhofer, T.; Monahov, I.; Matthes, F.: *Type-Safety in EA Model Analysis*. In *Enterprise Distributed Object Computing Conference Workshops (EDOCW), 2014 18th IEEE International, (submitted for publication)*. IEEE. 2014.

[Ro82] Robinson, J.: *Shedding darkness. Cambridge Journal of Economics*. 6(3):295–296. 1982.

[RWR06] Ross, J. W.; Weill, P.; Robertson, D. C.: *Enterprise Architecture as Strategy.*

Harvard Business School Press. Boston, MA, USA. 2006.

[Sc93] Scowen, R. S.: *Extended BNF – A generic base standard. Software Engineering Standards Symposium.* 1993.

[Sc99] Schneiderman, A. M.: *Why balanced scorecards fail. Journal of Strategic Performance Measurement.* 2(S 6):11. 1999.

[Sc01] Scott, M.: *Concepts of Programming Languages.* Morgan Kaufmann Publishers. 2001.

[Sc06] Schekkerman, J.: *Enterprise Architecture Score Card.* Technical report. Institute For Enterprise Architecture Development. 2006.

[Sc08] Schönherr, M.: *Towards a common terminology in the discipline of Enterprise Architecture.* In (Aier, S.; Johnson, P.; Schelp, J., Ed.): *Pre-Proceedings of the 3rd Workshop on Trends in Enterprise Architecture Research.* pages 107–123. Sydney, Australia. 2008.

[Sc09] Schwarzer, B.: *Einführung in das Enterprise Architecture Management: verstehen, planen, umsetzen.* Books on Demand GmbH. 2009. 9783837021226.

[Sc11] Schweda, C. M.: *Development of Organization-Specific Enterprise Architecture Modeling Languages Using Building Blocks.* Dissertation. Technische Universität München. München. 2011.

[Sc14] Schätzlein, M.: *A generic and integrated software support for the visualization of metrics in the context of EAM.* Master's thesis. Fakultät für Informatik, Technische Universität München. 2014.

[Se05] Sebis: *Enterprise Architecture Management Tool Survey 2005.* Technical report. Chair for Informatics 19 (sebis), Technische Universität München. Munich, Germany. 2005.

[Se14] Sebis: *Tricia version of the EAM KPI Catalog.* `http://131.159.30.135/pages/8wk3nt1cdms4/`. 2014.

[SK04] Schmid-Kleemann, M.: *Balanced Scorecard im IT-Controlling: Ein Konzept zur Operationalisierung der IT-Strategie bei Banken.* Treuhand-Kammer. 2004.

[SRS10] Siegwart, H.; Reinecke, S.; Sander, S.: *Kennzahlen für die Unternehmungsführung.* Haupt. 2010. 978-3-258-07485-6.

[St00] Sterman, J. D.: *Business dynamics: Systems thinking and modeling for a complex world.* Irwin McGraw-Hill. Boston [Mass.] and London. 2000. 0071179895.

[St01] Sterman, J. D.: *System Dynamics Modeling: TOOLS FOR LEARNING IN A COMPLEX WORLD. California management review.* 43(4). 2001.

[St06] Steinberg, R.: *Measuring ITIL: Measuring, Reporting and Modeling - the IT Service Management Metrics That Matter Most to IT Senior Executives.* Trafford Publishing. 2006. 9781412093927.

[St09a] Steinberg, D.; Budinsky, F.; Paternostro, M.; Merks, E.: *EMF – Eclipse Modeling Framework (2nd edition).* Addison-Wesley Longman. Amsterdam. 2009.

[St09b] Stutz, M.: *Kennzahlen für Unternehmensarchitekturen: Entwicklung einer Methode zum Aufbau eines Kennzahlensystems für die wertorientierte Steuerung der Veränderung von Unternehmensarchitekturen: Univ., Diss.–St. Gallen, 2009.* volume 31 of *Schriftenreihe Studien zur Wirtschaftsinformatik*. Kovac. Hamburg. 2009.

[St12] Strecker, S.; Frank, U.; Heise, D.; Kattenstroth, H.: *MetricM: a modeling method in support of the reflective design and use of performance measurement systems. Information Systems and e-Business Management.* 10(2):241–276. 2012.

[St14] Starke, G.: *Effektive Softwarearchitekturen: Ein praktischer Leitfaden.* Carl Hanser Verlag GmbH Co KG. 2014.

[Su13] Subashi, E.: *Establishing KPI systems for Enterprise Architectures: Risks and Countermeasures.* Master's thesis. Fakultät für Informatik, Technische Universität München. 2013.

[SWK13] Schütz, A.; Widjaja, T.; Kaiser, J.: *Complexity in Enterprise Architectures-Conceptualization and Introduction of a Measure from a System Theoretic Perspective.* Technical report. Darmstadt Technical University, Department of Business Administration, Economics and Law, Institute for Business Studies (BWL). 2013.

[SZ92] Sowa, J. F.; Zachman, J. A.: *Extending and Formalizing the Framework for Information Systems Architecture. IBM Systems Journal.* 31(3):590–616. 1992.

[Th09] The Open Group: *TOGAF, Version 9.1.* `http://www.opengroup.org/subjectareas/enterprise/togaf/` (cited 2014-05-05). 2009.

[Ti08] Tiarks, R.: *Software-monitoring mit hilfe eines dashboards.* Master's thesis. Universität Bremen. 2008.

[Tr10] Trumpfheller, J.: *Stefan Barenberg und Dr. Ute Lohse: Die Balanced Scorecard als ganzheitliches Performance Management-System in Finanzdienstleistungsunternehmen. Zeitschrift für die gesamte Versicherungswissenschaft.* 99(2):231–233. 2010.

[Tu01] Tufte, E. R.: *The visual display of quantitative informations 2nd ed.* Graphics Press. Cheshire and Conn. 2001. 0961392142.

[Tu05] Tuomela, T.-S.: *The interplay of different levers of control: a case study of introducing a new performance measurement system. Management Accounting Research.* 16(3):293–320. 2005.

[VRH04] Van Roy, P.; Haridi, S.: *Concepts, Techniques, and Models of Computer Programming.* MIT Press. 2004.

[Wi07a] Winter, R.; Bucher, T.; Fischer, R.; Kurpjuweit, S.: *Analysis and Application Scenarios of Enterprise Architecture–An Exploratory Study (Reprint). Journal of Enterprise Architecture.* 3(3):33–43. 2007.

[Wi07b] Wittenburg, A.: *Softwarekartographie: Modelle und Methoden zur systematischen Visualisierung von Anwendungslandschaften.* PhD thesis. Fakultät für Informatik, Technische Universität München, Germany. 2007.

[WRD03] Wallace, B.; Ross, A.; Davies, J. B.: *Applied hermeneutics and qualitative safety data: The CIRAS project. Human Relations.* 56(5):587–607. 2003.

Nomenclature

AL	application landscapes, page 122
AST	abstract syntax tree, page 131
BEAMS	Building Blocks for Enterprise Architecture Management Solutions, page 12
BI	business intelligence, page 153
BPMN	business process modeling notation, page 76
BSC	Balanced Scorecard Approach, page 17
CSS	Cascading Style Sheet, page 149
DSL	domain-specific modeling language, page 33
EA	Enterprise Architecture, page 1
EAMPC	EA Management Pattern Catalog, page 24
EBNF	Extended Backus-Naur Form, page 131
EMF	Eclipse Modeling Framework, page 31
GQM	goal-question-metric, page 76
IBBs	Information Building Blocks, page 13
IDE	integrated development environment, page 116
IS	information systems, page 6
IT	information technology, page 9
JSON	JavaScript Object Notation, page 125
KPI	key performance indicator, page 43
LBBs	Language Building Blocks, page 13
MBBs	Method Building Blocks, page 13
MMFS	Metric Management Fact Sheet, page 74
MVC	Model-View-Controller, page 110
MxL	Model-based Expression Language, page 115
OCL	object constraint language, page 31

OData	open data protocol, page 171
ROE	return on equity, page 16
SQL	Structured Query Language, page 116
TxL	Tricia implementation, page 115
UML	unified modeling language, page 31
VBBs	Viewpoint Building Blocks, page 13