Kapitel 1

Theorieverständnisse in Wirtschaftsinformatik und Betriebswirtschaftslehre

Stephan Zelewski, Naciye Akca und Malte L. Peters

Auf der Grundlage der zahlreichen Beiträge, die FRANK zu wissenschaftstheoretischen und auch wissenschaftssoziologischen Grundfragen der Wirtschaftsinformatik verfasst hat, wird das Theorieverständnis im „Mainstream“ von Wirtschaftsinformatik und Betriebswirtschaftslehre kritisch hinterfragt. Zur Verdeutlichung werden ein „schwaches“ und ein „starkes“ Theorieverständnis miteinander kontrastiert. Beide Theorieverständnisse werden im Rahmen des konventionellen Theorienkonzepts des „statement view“ erläutert. Zur Schärfung des „starken“ Theorieverständnisses, dem zufolge eine wissenschaftliche Theorie mindestens eine nicht-triviale nomische Hypothese umfassen muss, werden anhand des klassischen Transportkostenmodells die Schwierigkeiten aufgezeigt, die mit der Identifizierung einer solchen nomischen Hypothese verbunden sind. Zu diesem Zweck wird das klassische Transportkostenmodell als eine „Miniaturtheorie“ formalsprachlich rekonstruiert und erweitert. Die Argumentation wird in einen „Thesenstreit“ eingebettet, der von den Thesen der Theorielosigkeit und Theorieunmöglichkeit bis zur These einer starken nomisch-formalsprachlichen Theoriefundierung reicht.

1.1 Theorien in Wirtschaftsinformatik und Betriebswirtschaftslehre: ein kurzer Überblick

Zum Selbstverständnis sowohl der Wirtschaftswissenschaften im Allgemeinen als auch der Wirtschaftsinformatik und der Betriebswirtschaftslehre im Besonderen gehört oftmals der Anspruch, dass es sich um eine *theoretisch wohl-fundierte* Realwissenschaft handele.[1] Ähnlich häufig wird aber ebenso argumentiert, dass in den vorgenannten Realwissenschaften primär nicht auf eine theoretische Fundierung abgezielt werden solle, weil der Diskurs über

1 In diesem Beitrag wird – abgesehen von einigen Verweisen auf „ökonomische“ oder allgemein wirtschaftswissenschaftlich ausgerichtete Quellen – nur auf die Spezialfälle von Wirtschaftsinformatik und Betriebswirtschaftslehre als „verdeutlichende Beispiele“ explizit eingegangen. Dies ist schlicht dem Umstand geschuldet, dass sich die Verfasser vornehmlich im Bereich der Betriebswirtschaftslehre verorten (mit einer „interessierten Affinität“ zum Bereich der Wirtschaftsinformatik), sodass sie sich als nicht hinreichend kompetent fühlen, um sich zur theoretischen Fundierung der Volkswirtschaftslehre, des Wirtschaftsingenieurwesens und weiterer wirtschaftswissenschaftlicher „Bindestrich-Wissenschaften“ zu äußern.

Chapter 2

Academic Publishing and Academic Ethos – Looking Back to the Future

Stefan Klein

Ulrich and I shared an office for five years at GMD's Cologne Research Center on the Information Economy (1987–1992),[1] which laid the foundation for a lifelong friendship. We both had studied management at the University of Cologne (1978–1983) at a time when seminar theses and punch cards for the COBOL course as part of the IS specialization were written with. We compiled collections of index cards for referencing and abstracting literature and collected folders with copied articles and books. A notorious examination question in the oral IS exam was 'how many punch cards fit into this room?'. The historic reflections are obviously subjective and stylized, yet they try to capture the changing sentiments over the past decades as well as persistent issues like the role and ethos of science that are revisited and rearticulated over time. The disciplinary focus is on management, IS and computer science, other disciplines have different research and publication cultures. I use Elsevier as an example to illustrate the impact and risks of digitalization of academic publishing house, or 'information analytics businesses' as Elsevier has rebranded itself.

2.1 Utopia (Late 1980ies–mid 1990ies)

> *'Scientists and technologists, like all people who create, must dream, must put forth a vision of their future, or else they relegate their work to a kind of near-pointless incrementalism'* (Feigenbaum 1989, p. 118).

2.1.1 Information Search and Access

While working as PostDocs at GMD, we were discussing visions of digital research infrastructures. One example for a customized bibliographic database including links to the digital files of articles and books, i. e., software that supports the capturing of bibliographic information of articles or books based on standardized identifiers (today primarily DOI or ISBN), the assignment of keywords, annotations, the inclusion of digital files, referencing within word processors and the creation of bibliographies, which for the user – over time – becomes a private library accessible with a notebook or tablet. Such reference managers

1 Forschungsstelle für Informationswirtschaft which integrated research streams and researchers from GID, the Research Center for Information and Documentation `https://www.deutsche-digitale-bibliothek.de/item/FGOQHVT44GNYQCABC6EBCZ6AXXYJ7TBQ`

Chapter 3

Moral Responsibility in Conceptual Modeling

Alexander C. Bock and Jens Gulden

There is an irreducible ethical dimension to the activity of conceptual modeling. Conceptual models and conceptual modeling languages exert a profound influence upon our thinking and perception of real-world situations, and they govern the design of new artifacts. Ultimately, as Ulrich Frank has argued, they inform the development of possible future worlds. It is, therefore, imperative to attend to the many direct and indirect ways in which conceptual modeling is both the vehicle and the source of moral decisions. In this chapter, we reflect upon the notion of moral responsibility in conceptual modeling, which we take as an attitude of careful alertness to the ethical ramifications of both the activity and the products of conceptual modeling. Among the issues discussed are the contingency of the scope and content of conceptual models, the formative power and limits of modeling languages, the normative functions of conceptual models, and the role of conceptual modeling in the (re-)production of social reality itself. Our conclusions are intended to contribute to what might be developed into a body of guidelines for responsible conceptual modeling in the future.

3.1 Introduction

It is now almost half a century since *conceptual modeling* emerged as the distinctive modeling paradigm of the field of business information systems. At the time of its inception, conceptual modeling was primarily thought of as a means to support the design of computer-based information systems (IS), like database systems and object-oriented software systems (e. g., Brodie et al. 1984; Loucopoulos 1992; Rolland and Cauvet 1992). In the decades that followed, it gradually occurred to researchers and practitioners that conceptual models can be of value to many other areas of professional practice.

One result of this development is the field now known as *enterprise modeling*, which applies the principles of conceptual modeling to the integrated analysis and (re-)design of organizational structures, business processes, goal systems, and IS infrastructures (e. g., Zachman 1987; Frank 2002; Frank 2014a; Frank et al. 2014; Frank and Bock 2020a; Vernadat 2000; Sandkuhl et al. 2014).

Another product of the expansion of the use of conceptual modeling in recent decades is a sizable inventory of *domain-specific modeling languages* (DSMLs; e. g., France and Rumpe 2005; Frank 2013; Karagiannis et al. 2016). These languages provide modeling

Chapter 4

Conceptual Modeling: A Still Unfinished Saga

About Prejudices, Aberrations, Solutions and Challenges

Heinrich C. Mayr and Bernhard Thalheim

Conceptual modeling has a long and chequered history that goes far back into the past. In Computer Science, it has been discussed since the 70s of the last century, when – after some preceding work on data and process abstractions – the fundamental paper by Peter P.-S. Chen appeared in which he introduced the Entity-Relationship Model. In honor of the 65th birthday of Ulrich Frank who throughout his career has been deeply and also critically engaged with conceptual modeling, we present in this paper a few observations and thoughts on this history.

4.1 Introduction

This paper is intended as a little thank-you for Ulrich Frank, who has over many years decisively shaped the modeling scene, especially in Business Informatics, and surely will continue to do so. We would like to point out right away, however, that this is not a thoroughly serious, purely scientific paper. It is rather a collection of observations, questions, theses, and examples on the state of modeling which we have dealt with in the course of our many years of involvement with modeling ans partly also discussed in keynote speeches at various occasions. Accordingly, we make no claim to completeness: There are many things we do not mention, for example because we do not know anything about them, because we did not think they were worth mentioning, or because we simply forgot about them. Ulrich's 65th birthday, on which we extend our sincere congratulations, is a welcome occasion to present this thank-you, as he has contributed significantly to a better understanding and foundation of modeling, especially through his critical analyses but also with his constructive concepts. Modeling and modeling methods are crucial to most disciplines. We cannot do without them, even if they are not always appreciated, often underestimated, and in practice often dismissed with '*it's useless*'. After all, modeling has been widely researched, taught, and also practiced with a certain systematic for many years. However, many concepts are regularly 'reinvented' or renamed, many mistakes are made again and again, and prejudices are readily cultivated.

But there is progress: both in the theoretical foundation and in the realization of what is needed for and in practice. We will sketch a picture of this here – with a focus on more recent developments. Of course, this cannot be done without some recourse to the gray

Chapter 5

On Views, Diagrams, Programs, Animations, and Other Models

Hend*erik* A. Proper and Giancarlo Guizzardi

Humanity has long since used models in different shapes and forms to understand, redesign, communicate about, and shape, the world around us; including many different social, economic, biological, chemical, physical, and digital aspects. This has resulted in a wide range of *modeling practices.* When the models as used in such *modeling practices* have a key role to play in the activities in which these *modeling practices* are 'embedded', the need emerges to consider the effectiveness and efficiency of such processes, and speak about *modeling capabilities.* In the latter situation, it becomes relevant to develop a thorough understanding of the artifacts involved in the modeling practices/capabilities. One field in which models play (an increasingly) important role is the field of system development (including software engineering, information systems engineering, and enterprise design management). In this context, we come across notions, such as views, diagrams, programs, animations, specifications, etc. The aim of this paper is to take a fundamental look at these notions. In doing so, we will argue that these notions should actually be seen as specific kinds of models, albeit for fundamentally different purposes.

5.1 Introduction

Whenever we are confronted with complex phenomena, such as the processes we observe in nature, the construction of buildings, the design of information systems, etc, we tend to *'work with'* an *abstraction* (in our mind) of the actual phenomenon; zooming in on those *'properties'* of the phenomenon that matter to us, and filtering out all the properties that are not germane to the goals at hand. When we externalize this *abstraction* in terms of some artifact, then this artifact is a model (to us, as an individual) of the observed phenomenon.

More generally, one can observe how humanity has long since used models to understand, redesign, communicate about, and shape, the world around us, including many different social, economic, biological, chemical, physical, and digital aspects. These models may take different shapes and forms, such as sketches, precise drawings, textual specifications, or tangible forms mimicking key physical properties of some original. This wide spread, and natural (Zarwin et al. 2014) use of models has resulted in many different *modeling practices.*

When the models as created and/or used in such *modeling practices* have a key role to play in the activities in which these *modeling practices* are 'embedded', a natural need emerges to consider the effectiveness and efficiency of such processes, and speak about

Chapter 6

State Consistency – A Matter of Conceptual Modeling?

Elmar J. Sinz

Most conceptual models are expressed as data schemes, some as process schemes and only a few in other modeling languages. Modeling of state consistency, i. e., the goal-oriented alignment of the states of system components does not seem to play a role. However, the more the business world becomes distributed, cooperative, and volatile, this matter gets increasingly important. Has conceptual modeling a 'blind spot' here? The article shows the need for conceptual state consistency modeling. It is based on the concept of compensating transactions, borrowed from data management. Modeling of state consistency is demonstrated using the SOM approach.

6.1 Conceptual modeling and its 'blind spot'

Conceptual modeling has been an important topic in information systems engineering for decades (Frank et al. 2014). In particular, modeling of business reality is used here as a tool for analyzing and designing tasks and resources. The adjective conceptual indicates, on the one hand, that the focus here is on structures that are stable over the longer term and, on the other hand, that these can be seen independently of technical details.[1]. This allows to look at a conceptual model from two perspectives, a domain perspective, and an implementation perspective (Sinz 2021). The domain-oriented perspective describes the real-world context, while the implementation-oriented perspective describes its realization with IT.

Most conceptual models are expressed as data schemes; they capture aspects of the structure of a system. Sometimes process schemes are used, which describe behavioral properties. But what about state consistency, i. e., the goal-oriented alignment of the states of components of a distributed system? State integrity constraints (Schlageter and Stucky 1983, p. 291), formulated in a data schema by means of cardinalities of relationships, only capture the states within the respective system component, but not the overarching states of a distributed and volatile system. Does conceptual modeling have a 'blind spot' here?

One of the long-term development lines of our world is that production of goods and services increasingly takes place in distributed systems. Application systems (IT systems) have changed from monolithic blocks to highly distributed systems, consisting of a multitude of interacting components. The same is true for business operations. Hardly a production process is carried out by a single company; rather many larger and smaller companies are

1 https://www.umo.wiwi.uni-due.de/forschung/forschungsgebiete/konzeptuelle-modellierung/

Chapter 7

Epistemologically Motivated Modeling of Computer-Integrated Systems

Peter Fettke and Wolfgang Reisig

This paper frames and positions modeling in science and in engineering as an epistemological problem. Three requirements concerning structure, objects, and behavior that each modeling technique for computer-integrated systems must fulfill are identified and formulated as postulates. Corresponding modeling concepts are identified and epistemologically motivated. Furthermore, they are integrated into a comprehensive infrastructure for all basic tasks necessary to model computer-integrated systems.

7.1 Modeling in Science and Engineering as an Epistemological Problem

7.1.1 Modeling as an Epistemological Concept

Modeling in science and in engineering is characterized by models for *processes*, that is, the description of 'changes in time'. Usually, the real numbers are used as a time axis. The reason for the use of real numbers is the numerous properties of the modeled processes that can be derived from a model with the help of continuous functions, differential equations, and everything else known as 'the calculus'.

Informatics (or computer science) can be contrasted and characterized as the science of anything digital: Changes of facts (processes) are understood as discrete steps and not as continuous, gradual transitions. Facts of reality itself are represented symbolically. Is there a comparatively epistemologically well-founded, generic, comprehensive and outstanding modeling method for digital processes, analogous to the 'calculus' of the natural sciences, with deep-seated structural analysis methods? The remainder of this paper explains and motivates such a modeling method.

7.1.2 Modeling in Informatics from a Technical and a User's Point of View

Before going into details, we consider two approaches to the understanding of informatics as a science. Informatics emerged from the capabilities of *computational technology*. On a physical basis, computing technology offers a digital surface on which character strings are stored and transformed according to certain rules, and with the help of software. For this purpose, a multitude of programming and specification languages, databases, et cetera have been developed and used for decades. They are employed to meet specific aims, where the aims themselves remain outside the formal framework. All these concepts have a common epistemological basis: the theory of computable functions and anything typically subsumed

Chapter 8

Executable Multi-Level Modelling: Establishing Foundations, Methods and Tools

Tony Clark

System modelling has been a cornerstone of Information System Engineering for over 30 years. It aims to provide a basis for systematic representation, abstraction and analysis of system properties. Meta-modelling and Domain Specific Modelling both enhance system modelling through the use of abstraction at the type level. Until recently, there has been little attention to execution, methodology and structure at the type level of modelling. Ulrich Frank has made significant contributions to the field of *executable multi-level modelling* with the aim of addressing this gap. This paper reviews these contributions and discusses current and future challenges.

8.1 Introduction

System modelling has been a cornerstone of Information System Engineering for over 30 years. It aims to provide a basis for systematic representation, abstraction and analysis of system properties. Meta-modelling and Domain Specific Modelling both enhance system modelling through the use of abstraction at the type level. Until recently, there has been little attention to execution, methodology and structure at the type level of modelling. Ulrich Frank has made significant contributions to the field of *executable multi-level modelling* with the aim of addressing this gap. This chapter reviews these contributions and discusses current and future challenges.

The rest of this chapter is organised as a series of short sections introducing the key features of executable multi-level modelling with reference to the key contributions made by Ulrich Frank. The chapter is not intended to be self contained, nor will it contrast the contributions with those of other researchers (some of whom present significantly different approaches to MLM). It represents a review of the key contributions of a major contributor to an important field with links to publications for the interested reader to explore the details.

8.2 Multi-Level Modelling

The field of Multi-Level Modelling (MLM) can be traced back over 20 years (Atkinson and Kühne 2001) where it was introduced to address shortcomings in standard approaches to modelling as exemplified by the Unified Modelling Language (UML).

Chapter 9

Projective Multi-Level Enterprise Architecture Modeling with MEMO

Colin Atkinson, Christian Tunjic and Arne Lange

At the heart of the current drive for ubiquitous digitisation and pervasive cyber-physical systems are enterprise architectures that comprehensively describe the interplay between organisations' goals, plans, and IT solutions. However, traditional enterprise architecture modelling approaches are not fit for purpose since they are unable to cope with the scales and dynamics of modern enterprises and systems. This paper explores how the full and synergistic integration of three forward-looking modelling principles can help address this problem and provide a sound foundation for the enterprise architecture models needed in the future.

9.1 Introduction

To cope with the increasing size, complexity, and scope of Enterprise Architectures (EAs), there is a growing consensus that Enterprise Architecture Modeling (EAM) environments need to be built on three core principles. The first is that they should be fundamentally *view-centric* so that their many stakeholders can be provided with precisely the information they need at the right time and in the right form (Frank 2002). For this reason, EAM approaches, from Zachman (1987) onward, have been structured around a viewpoint framework that describes what view types are available, from what perspectives (i. e., viewpoints), and how they are organised. The second is the principle that EAM approaches need to be founded on *multi-level modelling* approaches rather than on the traditional two-level modelling approaches supported by the UML and most other EAM languages (Frank 2014b). Two-level approaches struggle to adequately support the deep characterisation patterns frequently occurring in EAs or the flexible definition of new languages for new view types. Moreover, when the scope of EAMs includes the 'run-time' phase of an enterprise's operation, multi-level modelling approaches allow all required information (or data), at all levels of abstraction to be represented in a seamless and unified way. The third is the principle that EAM environments need to be fundamentally *'projective'* (ISO/IEC/IEEE 2011), in the sense that all views seen by stakeholders should be explicitly projected on demand from a Single Underlying Model (SUM) (Atkinson et al. 2010) using model transformation defined by end users. The alternative 'synthetic' approach, in which views actually hold primary information and need to be kept mutually consistent by a multitude of pairwise consistency-preservation rules, simply does not scale up to large EAMs.

Chapter 10

The Product Perspective in Multi-Perspective Enterprise Modelling

What Information about Products is Relevant for Enterprise Models?

Kurt Sandkuhl

Most enterprise modelling approaches use multiple perspectives to provide guidance in the modelling process and to improve manageability of the resulting enterprise model. The perspectives are supposed to address particular concerns of an enterprise and to provide adequate description and modelling practices. The focus of this paper is on the product perspective, i. e., on constructs, structures and rules for modelling products and services of an enterprise. The contributions of the paper are (a) an analysis of product information relevant for enterprise modelling, (b) an overview to prominent enterprise modelling approaches and how they support product modelling, and (c) selected practices for product modelling. We argue that the trend of digital transformation of enterprises motivates more support for product modelling in the context of enterprise modelling.

10.1 Introduction

Many enterprise modelling (EM) approaches use multiple perspectives when analysing the current situation of an organisation or designing the future situation. The motivation for multiple perspectives in general is to reduce complexity of the modelling process and to improve manageability of the resulting enterprise model (cf. Section 10.3.1). Each perspective is supposed to address particular concerns of an enterprise and to provide adequate constructs, structures and rules. This supports modellers in understanding, analysing, capturing and representing what is relevant for different groups of stakeholders and/or different modelling purposes (Frank 2014). Examples of perspectives often visible in enterprise modelling approaches are the process perspective (i. e., the work processes of an enterprise with their decomposition in tasks, information flows and control flows) or the organisation structure perspective (i. e., the structure of organisation units with decision structures, roles and positions). The focus of this paper is on a perspective that is not explicitly mentioned in most EM approaches: the product perspective, i. e., the products offered by an enterprise with decomposition structure, features, target groups, and other product-related information.

From an economic viewpoint, the purpose of an enterprise is commonly the creation of value for its customers and profits for its owners by providing defined products or services meeting the customers' needs and fulfilling regulatory, environmental, ethical, and market standards. Consequently, a substantial part of an enterprise's operation has to focus on

Chapter 11

Bridging the Mental and the Physical World: Conceptual Modeling and Augmented Reality

Hans-Georg Fill and Fabian Muff

Whereas conceptual modeling is today widely used for representing knowledge for the purpose of communication and understanding, the combination with augmented reality technologies permits for the first time to anchor this knowledge formally to objects in the physical world using electronic means. In addition, conceptual modeling may help to support the design of complex augmented reality applications and thus enable non-technical users to better engage with this technology. In this chapter, we thus explore the combination of conceptual modeling and augmented reality by focusing on the role of the subject and how its perception is augmented using augmented reality technologies. From this, we derive two directions in the form of a. *Augmented Reality-based Metamodeling and Modeling*, and b. *Knowledge-based Augmented Reality* and illustrate them with recent examples.

11.1 Introduction

The benefits of *models* are today widely known and largely undisputed. One of the probably most known examples includes architectural models that are an indispensable prerequisite for constructing houses (Zachman 1987). Thus, models not only stand for the *artistic* creation of an artifact (Thalheim 2012b). They also constitute a representation of the *knowledge* that is required for assembling a building's components in such a way that it meets the desired properties of its future owner (Karagiannis et al. 2017; Abazi et al. 2011; Johannsen and Fill 2015), e. g., in terms of stability, energy efficiency, comfort, or legal and safety requirements. For achieving this kind of knowledge representation, we will focus our discussion on the field of *conceptual modeling* that requires the application of a modeling language, which pre-defines the concepts to be used in models (Harel and Rumpe 2004; Thalheim 2012a; Stachowiak 1973). Conceptual modeling plays an important role in the fields of Computer Science and Business Informatics (Härer and Fill 2020), where it supports for example the elicitation of software system requirements, e. g., Yu et al. (2011), the creation of databases, e. g., Glässner et al. (2017), or the management of business processes and IT systems, e. g., Dumas et al. (2018) and Moser et al. (2022).

However, the adoption of conceptual modeling and its application in companies in the form of enterprise models is characterized by several challenges as recently laid out in Sandkuhl et al. (2016), Sandkuhl et al. (2018) and Frank (2014). Although conceptual

Chapter 12

Towards Tailored Support for Design Practices in Collaborative Modeling Sessions

Tobias Kautz and Robert Winter

Especially in early phases of digital business innovation endeavors, collaborative design sessions are critical for creating a shared understanding, aligning diverse perspectives and objectives, and making fundamental design decisions. As knowledge about the characteristics of these sessions is scarce, existing support artefacts (e. g., methods, techniques, or software tools) cannot be easily selected, let alone tailored to the specific characteristics of such a session resulting in only generic guidance and missed effectivity potentials. This paper reports intermediate results of a design science research project aimed at developing tailored support artefacts for such sessions. As a foundation to achieve higher effectivity, a theory-informed conceptual model is proposed from which the classification dimensions for collaborative design sessions are derived. Classification dimensions focus on involved stakeholders, topics, i. e., compositions of focal entity types, and information characteristics derived from abstraction principles. Based on interviews and surveys, fifteen session types along four phases are identified. A concrete plan for future research is provided to investigate the explored session types and develop tailored support artefacts for them.

12.1 Introduction

For years and most recently intensified through the Covid-19 pandemic, organizations are designing and implementing, sometimes voluntarily, sometimes not, digital technologies. These change endeavors impact, e. g., customer behaviors, value propositions, value chains, and sometimes entire business models (Dąbrowska et al. 2022). On an organizational level, we will refer to them as *digital business innovation* (DBI) endeavors. DBI endeavors can be focused on a specific business unit (e. g., a certain component of the product and service portfolio), but also across certain business units or ultimately on the entire enterprise. The latter type of DBI endeavors is also known as digital (enterprise) transformation. Depending on its focus and also the specific wording used in the respective enterprise, a DBI endeavor can be organized as a project, a program or even a strategic initiative (Klingebiel and De Meyer 2013; Pellegrinelli 2011). However, even when implemented in one business unit only, the value proposition and IT landscape changing character has consequences for the entire enterprise. Even the introduction of a single new service may compete with

Chapter 13

Omnichannel Processes in Retailing SMEs: Digitalisation in a Furniture Company

Jörg Becker, Paul Kruse, and Andreas Hermann

13.1 The Starting Point of Digital Transformation

Digital transformation is a complex endeavour companies face to stay competitive in an ever-changing business environment (Fletcher and Griffiths 2020). Various options are available to digitalise a company through the continuous development of new digital technologies. However, to implement the new technology, the processes within the organisation need to be transformed as well. Hence, many scholars develop procedure models structuring the general transformation process (Barann et al. 2019). To apply the digital transformation successfully, general knowledge about the existing process landscape is required. With this process knowledge, the company's digital transformation will achieve the desired result. Unfortunately, the general process knowledge in small and medium-sized enterprises (SMEs) often needs to be improved (Mendling et al. 2010). As a consequence, SMEs often do not know the interrelation of their processes and rather conduct an unstructured transformation process.

To conduct a structured transformation, SMEs need good Business Process Management (BPM) practices. An overview of the existing processes is required to set up BPM successfully. A rigorous process model can provide this overview. As models are an abstraction of reality, the respective process model depicts a real-life process in an abstract form. Without the process model, employees know their daily routines and often do not have the bigger picture in mind. Collecting and synthesising the individual processes into an exhaustive process model can provide a general overview of joining the dots. Yet, achieving this overview requires a process-oriented mindset and appropriate modelling skills. However, many different modelling techniques consist of different modelling layers and loose or fixed levels of detail. Hence, a general understanding of the respective modelling technique is required. The required knowledge to model processes correctly is often lacking in SME as they often do not have the resources or personnel to apply proper process modelling within the organisation (Mendling et al. 2010). Next, external experts, such as consulting companies, are often too expensive for SMEs. Even though a consulting company conducted the modelling, the employees of the SME need additional training to read and understand the respective models. Summarising, a rigorous BPM is based on well-modelled business processes which require expert knowledge and is often unnecessarily complex.

Chapter 14

Towards Reference Models with Conformance Relations for Structure

Marco Konersmann, Judith Michael and Bernhard Rumpe

The term 'reference model' is broadly used in publications and discussions. To our understanding, the role of reference models and their use in modeling is not standardized and understood mostly informally in many communities. We posit that a more concise and formal understanding of the concept of a reference model is needed, to help make tool-assisted use of reference models and their relation to concrete models. In this contribution, we present our understanding of reference models and conformance relations, which specify which models are valid concretizations of a reference model. We argue why no general conformance relation can exist for all modeling languages. On the example of data models using UML class diagrams, we discuss potential recurring requirements of conformance relations, representations for mappings between concrete models and reference models, and the challenge of given code implementations to be attached to reference models and how to transfer this code to a concretization.

14.1 Introduction

In existing publications, there seems to be no agreement on what the term 'reference model' explicitly means. Authors define 'technical reference models' (Joshi and Michel 2008), 'business reference models' (Eom and Fountain 2013; Frank and Strecker 2007; Frank 2007), 'business process reference models' (Scheer 1994; Fettke et al. 2006; Reinhartz-Berger et al. 2010), the 'ISO/OSI 7 layer communication reference model' (American National Standards Institute 1981), 'enterprise architecture reference models' (Zimmermann et al. 2015), 'performance reference models' (Forme et al. 2007), 'reference models for deep learning frameworks' (Atouani et al. 2021), 'reference models for digital shadows' (Michael et al. 2023), and various others. Occasionally meta-models (Bucaioni et al. 2022; Michael et al. 2023; Mayr et al. 2018), meta-models and OCL (van der Aalst and Kumar 2001), or component-connector architecture models (Dalibor et al. 2022) are used and treated as reference models.

Gray and Rumpe raise the question of whether it is possible to leverage the notion of a reference model. In particular: 'How can the notion of a reference model, explicitly denoted in a given modeling language, be formalized together with a precisely defined and tool-assisted notion of a conformance relation that would enable concrete realizations of the reference model?' (Gray and Rumpe 2021). In this contribution, we present a formalization of reference models for structure data and their conformance relations along with operations

Chapter 15

Enterprise Use Cases Combining Knowledge Graphs and Natural Language Processing

Phillip Schneider, Tim Schopf, Juraj Vladika and Florian Matthes

Knowledge management is a critical challenge for enterprises in today's digital world, as the volume and complexity of data being generated and collected continue to grow incessantly. Knowledge graphs (KG) emerged as a promising solution to this problem by providing a flexible, scalable, and semantically rich way to organize and make sense of data. This paper builds upon a recent survey of the research literature on combining KGs and Natural Language Processing (NLP). Based on selected application scenarios from enterprise context, we discuss synergies that result from such a combination. We cover various approaches from the three core areas of KG construction, reasoning as well as KG-based NLP tasks. In addition to explaining innovative enterprise use cases, we assess their maturity in terms of practical applicability and conclude with an outlook on emergent application areas for the future.

15.1 Introduction

As modern organizations continuously adapt to the evolving requirements of the digital age, the importance of successfully managing enterprise data has never been greater. In this article, we define the term 'enterprise' as a large-scale business, which operates on a national or international level and typically involves significant risks and resources. The competitive advantage of data-driven decisions affects all industry sectors and nearly every part of the value chain (Schopf et al. 2022b). For example, market trends are analyzed for business development, production is optimized through process metrics, and customer reviews are monitored for predictive maintenance. However, raw data alone is insufficient for decision-making. In order to become actionable information, data has to be endowed with meaning and purpose (Rowley 2007). This can be achieved by data enrichment through a relevant context. A compelling approach to achieve this is by modeling knowledge in the form of graph connections between data items (Martin et al. 2021).

In view of the above, knowledge graphs (KGs) have emerged as a powerful representation for integrating knowledge from multiple information sources. They model semantic relationships among key entities of interest, such as customers, markets, or services. In this sense, a KG serves as an abstraction layer to combine both business data and explicit business knowledge. Even though Google coined the term knowledge graph back in 2012 when they announced their new web search strategy (Singhal 2012), it is not an entirely

Chapter 16

AdoScript: A DSL for Developing Conceptual Modeling Methods

Dimitris Karagiannis

To support modeling method engineers, this work adopts the paradigm of domain-specific languages and proposes a specialization of it labelled as 'domain-specific language for modeling method realization' (MM-DSL), which is demonstrated by the instance of AdoScript – a language that has been the key enabler for numerous modeling methods and tools developed over the years. An MM-DSL builds on foundational meta-concepts and answers requirements that are specific to the domain of conceptual modeling. This chapter presents the modeling method engineering language AdoScript and how it meets those requirements. It is complemented by the more abstract languages and formalisms MetaMorph and FDMM which serve for the definition and specification of modeling methods according to the Generic Modeling Method Framework (GMMF). Together, this trilogy of languages and their key meta-concepts are operationalized on a concrete metamodeling platform, ADOxx.

16.1 Introduction

To push forward a scientific community like the one of Conceptual Modeling, it is not only necessary but existential to have also understandable, usable, and applicable standards and software platforms. Conceptual Modeling is closely related to disciplines like Software Engineering, Web Design, and Cloud Computing, which have specific standards like MOF, HTML, and ArchiMate. While Conceptual Modeling has previously adopted standards like ER and BPMN for modeling perspectives, it currently lacks standards for language engineering, similar to HTML, and development platforms like Eclipse.

For this reason, we propose in this chapter a starting point targeting a standard for modeling method engineering. This should serve as invariant over time, of course with adaptions according to technological progress regarding the concrete means of realization and operationalization. The chapter introduces AdoScript, a DSL for realizing modeling methods (i. e., a MM-DSL) that encompasses those invariant aspects and meets the requirements pertaining to them. The key meta-concepts fundamental for the proposed language will be examined; they can be interpreted as requirements for a MM-DSL that targets productivity benefits for modeling method implementation, the typical result being a conceptual modeling tool.

In Section 16.2, we give an overview of the relevant notions like domain-specific conceptual modeling and modeling method engineering. The subsequent Section 16.3 then

Chapter 17

Model-Driven Development: From Strategy to Code

Oscar Pastor, Rene Noel, Rosa Velasquez and José Ignacio Panach

Model-driven development puts software abstractions at the centre of the software development process. Through model-to-model transformations, business knowledge abstractions help developers to align software design with business requirements. However, the constantly changing environment and the social, cross-disciplinary nature of software development teams sets two main challenges to model-driven methods. On the one hand, modern organisations have an accelerated approach to defining, deploying, and measuring the success of business strategy plans, making them a central part of the software development process. On the other hand, business users need a shared comprehension of business processes, which might be enabled only by understandable business process models. This chapter presents a model-driven method that starts by representing the business strategy that triggers the software development initiative, then provides an assisted business process modelling method to improve models' understandability, and then connects with a sound model-driven method that generates the software code. The method aims to help design understandable business processes and software components aligned to business strategy and improve the traceability of the whole development process from strategy to code.

17.1 Introduction

Model-Driven Development (MDD) is the systematic use of software abstractions (models) as primary artefacts during a Software Engineering (SE) process (Mellor et al. 2003). Following the Model Driven Architecture (MDA) (The Object Management Group 2014) approach, software models can be connected with different system and business abstraction levels. The abstraction levels are connected through model-to-model transformations from business information at the Computation Independent Modelling (CIM) level, which can then be transformed into Platform Independent Models (PIM) that embody the design of the system and then into Platform Specific Models (PSM) with technical details to support the automatic generation of the system code.

One of the key advantages of the MDA-based methods is to allow business stakeholders from different domains to participate in the development process (The Object Management Group 2014). However, in model-driven processes, all the stakeholders must understand the models. The understandability of conceptual models is an open research problem (Houy

Kapitel 18

Die Entdeckung des Plurals für die Programmierung

Friedrich Steimann

Während in natürlicher Sprache Singular und Plural gleichberechtigt nebeneinander stehen, gibt es in Programmiersprachen praktisch nur den Singular: Wo viele vorkommen, werden diese zu einem (einer Collection oder ähnlichem) verpackt. Zwar reduziert diese Reifizierung von vielen zu einem die Zahl der benötigten Sprachmittel, jedoch macht sie die Programme, die viele handhaben müssen (und das sind praktisch *alle* Programme) unnötig komplex. Mit meinem Beitrag zeige ich, dass eine Einführung des Plurals in die Programmierung nicht nur ökonomisch, sondern auch wohlfundiert sein kann. Zugleich mache ich deutlich, wie sie die Programmierung näher an die Modellierung heranrückt, in der der Plural schon immer zum Selbstverständnis gehört.

18.1 Einleitung

Alles ist ein Objekt. Diesem Leitsatz der objektorientierten Programmierung werden Anhänger der funktionalen Programmierung entgegenhalten, dass in Wahrheit alles eine Funktion ist, aber der entsprechende Paradigmenstreit soll – zumindest in diesem Beitrag – nicht weiter interessieren. Vielmehr will ich mich daran abarbeiten, dass in beiden Sätzen „alles" über Einzelne rangiert, wie man am Singular der Kopula „ist" ablesen kann. Obwohl „alles" eigentlich allumfassend ist, schließt es Viele (Plurale) aus: „Zwei Objekte ist ein Objekt" wäre grammatikalisch falsch und „zwei Objekte sind ein Objekt" wäre es inhaltlich.

Was man sprachlich vielleicht als Spitzfindigkeit abtun könnte, ist programmiertechnisch Dogma: Während es in einem Programm zwar in der Regel viele Objekte (und viele Funktionen) gibt, so kann an jeder Stelle des Programms (also dem, was mit „alles" gemeint ist) immer nur ein Objekt stehen — sollen es viele sein, müssen sie für diesen Zweck durch ein Objekt repräsentiert werden. So gesehen sind eben auch viele Objekte ein Objekt (ein Vieles), so dass der Satz „alles ist ein Objekt" durch den Einschluss Vieler unter „alles" nicht ungültig wird. Für Funktionen gilt ähnliches.

Die logische Zusammenfassung von vielen[1] zu Einem hat viele Vorteile. Historisch gesehen hat sie sogar einen Umbruch erheblicher Tragweite bewirkt: Mit Georg Cantors Einführung von Mengen als „jedes Viele, welches sich als Eines denken lässt" (Cantor 1883, S. 587) ist die Mathematik neu begründet worden. So lassen sich Zahlen genauso als Mengen

1 Ich schreibe hier „viele" klein, da ich nicht den Plural von einem Vielen (also mehrere Viele) meine, sondern mehrere Einzelne.

Chapter 19

Steps Toward Articulating a Work System Perspective that Addresses a Grand Challenge for the IS Discipline

Steven Alter

This chapter is a contribution to an anthology in honor of Ulrich Frank. It describes some of the twists and turns in my research both before and after my 2017 visits to Ulrich's Enterprise Modelling Research Group at the University of Duisburg-Essen. Those visits helped me rethink aspects of personal research goals and directions related to one of the grand challenges for IS research.

19.1 Introduction

My visits to Ulrich's group helped me rethink aspects of personal research goals and directions related to one of the grand challenges for IS research that were identified by a Delphi study involving 143 IS academics 35 years after the first ICIS meeting (Becker et al. 2015). That challenge, 'rethink the theoretical foundations of the IS discipline,' was tied for first of 21 in being viewed as a grand challenge (74.5 % yes, 14.5 % no). It was ranked third of 21 as having an impact on the discipline if it were solved, and was ranked 13 of 21 in terms of time frame, i. e., whether it could be dealt with or solved within 10 years. I think that the work system perspective that I have developed gradually over much of my academic career addresses important parts of that challenge, as will be discussed here.

Upon reading the grand challenge article around four years ago, I was surprised that the topic of theoretical foundations was viewed as more of a grand challenge than 'developing model–driven methods and tools for the full–scale automated generation of implementation–ready IS,' which was seen as a grand challenge by only a minority of the Delphi panel (23.6 % yes, 61.8 % no). In contrast, I saw that challenge as a more difficult and consequential challenge based on my minor involvement in an automatic programming project while I was a PhD candidate. I think that challenge comes closest to categorizing the research performed by Ulrich and his Group.

I was lucky to meet Ulrich Frank around 10 years ago and to discover that we could have valuable conversations even though we pursued different research goals in quite different ways. I was trying to develop ideas and methods that could be useful in understanding systems in organizations while he was trying to develop methods and tools for automated generation of implementation-ready IS. My visit in May 2017 involved teaching a course for UDE's Visiting Scholar Academy, where the instructor teaches a short course on topics that might not otherwise be covered. The second visit was a personal attempt to do something useful during part of the time between the PoEM (Practice of Enterprise Modeling) conference

Chapter 20

What a Wonderful Modeling World! Tackling Conceptual Diversity in Enterprise and Information Systems Modeling

Marc Frerichs and Markus Nüttgens

Starting with flow process charts, science and practice have developed a large and complex modeling landscape over the last 100 years based on a variety of modeling scenarios with an almost unmanageable number of generalistic or domain-specific methods. The resulting conceptual diversity has led to the existence of modeling methods for nearly every conceivable use case. Modeling, which is supposed to make complexity controllable through structure, has itself become a complex world. The selection of a modeling method optimal for the use case is difficult today. Due to the variety of business processes, the situation is aggravated by the circumstance that, depending on the use case, different modeling techniques are often used, resulting in inflexibility. In this chapter, we conduct a literature review on diversity in the area of conceptual modeling, and specifically in enterprise and process modeling, with the primary question of which theories, methods, and tools exist to describe the modeling landscape's diversity and make it manageable. The results of our work show that there have been different approaches over the last decades (meta modeling, standardization, and model transformation), but none of them could really prevail. As a thought-provoking teaser, we propose an approach for an independent middleware layer (process model warehouse concept) that can extract data from (enterprise) systems and enrich it using linked Business Process Management (BPM) tools and vice versa.

20.1 Introduction

'You have a character that's all of a piece, and you want the whole of life to be of a piece too—but that's not how it is. [...] All the diversity, all the charm, all the beauty of life is made up of light and shade' was one of Oblonsky's nuggets of wisdom addressed to his friend Levin in Tolstoy's work Anna Karenina. Indeed, the world is complex, diverse, full of uncertainties and it's definitely not 'all of a piece.' The interplay of things – as parts of a bigger whole – is what makes the world so exciting, but also hard to describe. The intention to master this complexity led to the development of various modeling approaches.

Models, as abstract representations of real-world objects, go back to the Stone Age. In anthropology, the ability to abstract and build models is seen as a competitive advantage of human beings. Schichl (2004) summed up the long history of modeling, providing a

Chapter 21

The Process of Detecting Suspicious Communication with Psycholinguistics and Text Mining Techniques

Ewelina Księżniak and Witold Abramowicz

This chapter presents the process of detecting suspicious communication with text mining and psycholinguistics techniques. The chapter describes the key finding of research in the field of detecting deception in text with psycholinguistics and presents the algorithm designed to detect communication aims to deceive or harm the receivers of the messages.

21.1 Introduction

We discuss the process of detecting suspicious communication with text mining and psycholinguistics analysis techniques. According to many researchers lying is a cognitively complex task that influences the language used by liars. Therefore, it is possible to detect deception in a text by linguistics analysis. The proposed method combines the discoveries from the psycholinguistics and computer science fields and is applicable in many areas. The term 'suspicious communication' may be understood as any communication between individuals or business parties that aims to deceive or harm the receiver of the message.

In the chapter, we focus on presenting the application of the proposed algorithm to detect financial scams with an example of the Enron case study. However, the discussed method may also be applied to detect fake news or phishing. The chapter is divided into three main sections:

- Psycholinguistics in deception detection – this section discusses key concepts related to deception detection with psycholinguistics (Section 21.2).

- The process of detecting suspicious communication with Enron emails case study – in this section, we describe the proposed method of detecting suspicious communication with psycholinguistics with the Enron emails case study. We also briefly discuss Enron's history and present the data cleaning process (Section 21.3).

- Other applications and future work – this section examines future prospects and summarizes the potential use of the algorithm in various fields, including fake news detection (Section 21.4).

Chapter 22

The Next Generation of ERP Systems: Problems of Traditional ERP-Systems and the Next Wave of Really Standardized ERP Systems

Reinhard Schütte

The research work and publications on ERP systems in the IS community are multi-layered, whereby here the banal attempt of a keyword-based literature analysis shan't be undertaken, being as insightful as a forest rich in sunshine. It is the peculiarity of a Festschrift that scientific freedoms present itself to the authors in dealing with the topic, which are closed in the other scientific publication practice. These freedoms enable the author to work on a topic where access to the subject matter is more difficult, in this case enterprise-wide standard software in general and ERP systems in particular.

22.1 Status Quo of ERP Systems

22.1.1 Introduction

There is a wide range of contributions dealing with teaching and research aspects of ERP systems (for an overview, see Bahssas et al. 2015 and Klaus et al. 2000), however, the question arises whether the nexus to 'ERP systems in enterprises' is sufficiently addressed. Based on more than twenty years of experience in the development and implementation of standard software systems, the author asserts that this is indeed not the case. There is a lack of work dealing with the technical basis of standard software and the interests of standard software producers on the one hand and the introduction and use of standard systems in companies depending on this technical basis on the other hand. SAP software in its different generations from R/2, R/3 up to S/4 HANA can be understood as the prototype of enterprise-wide standard software. Even in its current version, the software comprises up to 400 million lines of code; the reengineering of the old R/3 coding announced by Plattner and Leukert (2015) has failed to materialize, according to the author's assessment.

Accommodating individualizations is a considerable challenge in terms of software technology. Thus, not only the standard, provided upon delivery of the system, is relevant for decision-making from a business point of view, but also the technical capabilities for enhancing and adapting it. This technical environment then becomes even more important when implementing updates from the software manufacturer while maintaining formerly adopted individual developments. In an EDEKA project known to the author, a standard system now has about 4 million lines of code of additional developments that would not have been safely manageable for so long without the features of SAP software logistics

Chapter 23

Process Management: A Reflection from a Practical Perspective

Stefan Jablonski and Stefan Schönig

Process management is an already established method and technology for enacting processes in organizations. Both, academia and professional practice, are subscribing to this statement for certain. Nevertheless, it is very revealing to recall the last years of research and practical usage of process management approaches and draw conclusions from that. This is why we like to share our fragmental and very personalized insights into research on the one hand side and professional practice on the other hand side. We intend to motivate both academia and professional practice to further develop and enact process management concepts since we are firmly believing in this approach and would like to see this domain emerging further. Especially, we like to foster a broader dialogue between these two disciplines.

23.1 Introduction

Process management is a widely accepted approach to organize collaboration between distributed process participants. Since about three decades there are enormous research activities in that domain. Also practical systems have been evolving. Nevertheless, we have experienced a gap between concept and method development in the research domain and enactment of those ideas in praxis. In this discourse we like to reveal and discuss some of those gaps. Thereby, we are pursuing two perspectives. On the one hand, we especially look into the modeling phase of a process application, whereby we do not neglect the other phases of a process lifecycle completely. On the other hand, we adopt the perspective of a software engineer that has the obligation to enact a process management application in an application-oriented setting. Finally, our first goal is to stimulate vendors of process management tools to more enrich their systems by integrating practical and useful research concepts. We do this since we know from our academic background that there are powerful and interesting concepts already developed that are still not available in practical tools but are extremely helpful when enacting practical process applications. Since not all aspects of process management are already researched perfectly, our second goal is to motivate researchers to continue research on these still open issues and not to neglect them because they are not most current.

We structure our discourse into two main sections. In Section 23.2, the fundamentals of process management are introduced in a nutshell, only just to sketch the main characteristics and challenges of process management. Those issues are necessary to know in order to