

A Comparative Analysis of Selected Enterprise Modeling Approaches

Alexander Bock, Monika Kaczmarek, Sietse Overbeek, and Michael Heß

Chair of Information Systems and Enterprise Modelling
Institute for Computer Science and Business Information Systems (ICB)
Faculty of Business Administration and Economics, University of Duisburg-Essen,
Universitätsstraße 9, 45141 Essen, Germany
{alexander.bock,monika.kaczmarek,sietse.overbeek,m.hess}@uni-due.de

Abstract. Complexity inherent to the management of organizational action recommends the use of instruments that support the structured description and analysis of organizations. A variety of enterprise modeling (EM) methods have been developed to serve these purposes. To contribute to the elucidation of their conceptual differences, overlaps, and focal points, this paper analyzes four selected EM methods based on a designed analysis framework. It includes an assessment of the methods' key goals and purposes, central assumptions, and concepts. The paper concludes with a suggestion of future research topics.

Keywords: Enterprise modeling method, comparative analysis.

1 Introduction

Enterprise modeling (EM) is commonly regarded as the construction and use of conceptual models to describe, analyze, and (re-)design different aspects of an organization (e.g., [1, pp. 942-943], [2, p. 1]). Enterprise models are built from modeling concepts [3, p. 251], defined in modeling languages, which constitute abstractions of organizational aspects (organizational action systems) and information systems (IS) [1, p. 942]. Examples of enterprise modeling methods include ArchiMate [4], Architecture of Integrated Information Systems (ARIS) [5], Business Engineering (BE) [6], Design and Engineering Methodology for Organizations (DEMO) [7,8], For Enterprise Modeling Method (4EM; formerly Enterprise Knowledge Development) [9,10], Multi-Perspective Enterprise Modeling (MEMO) [1,11,12], Semantic Object Model (SOM) [13], The Open Group Architecture Framework (TOGAF) [14], and Work System Theory (WST) [15].

Conceptual overlaps between these methods can be assumed, as in general, they all aim at the development of structured descriptions of enterprises. The methods are not identical, though, as they emerged from different backgrounds, were developed with different purposes in mind, and are based on different assumptions. As each method, moreover, is using its own terminology, similarities and distinctions between the methods are not obvious. Assessing the methods

with respect to the requirements of different application contexts becomes a challenge, and integration benefits for both the usage and evolution of EM methods may remain unexploited. To contribute to the consolidation, integration, and evolution of EM methods, and, possibly, to advance the field of EM, a need emerges to comparatively analyze existing EM methods [16, p. 1-3], [3, p. 250].

The main goal of this paper is to analyze a set of EM methods to contribute to understanding the range of conceptual and foundational similarities and differences between existing EM methods. The analysis aims to answer the following questions: What are the main similarities and distinctive features of the selected EM methods? What are their focal points? Finally, which areas could be further investigated in order to potentially advance research and practice of EM?

In order to answer these questions, we devise an analysis framework and conduct a comparative analysis that places focus on the identification of the essential characteristics of a limited set of EM methods. According to a scheme proposed by [16, p. 111], we conduct a ‘vertically dominant’ analysis. As an exemplary selection, four EM methods are chosen for our analysis: ArchiMate, DEMO, MEMO, and WST. These methods are deliberately selected such as to achieve a variety in background, underlying theoretical references, and key application scenarios. Further, already performed comparisons (e.g., between ARIS, BE and SOM [17]) have been taken into account, and only more mature approaches being in the development for at least one decade have been considered.

The paper is structured as follows. First, related work is shortly described in Sect. 2. In Sect. 3, the utilized analysis framework is presented. A conceptual analysis and comparison of the methods based on the defined framework is conducted in Sect. 4. The results are discussed in Sect. 5. Concluding remarks and an outlook on possible future research directions are provided in Sect. 6.

2 Related Work

The comparison of alternative modeling methods (i.e., of “alternative suggestions to (re-)construct the world” [18]) is an important challenge in the IS field. This is reflected in related work in two ways. Firstly, there is literature concerned with the approaches to evaluating modeling methods. Secondly, there is a range of work conducting concrete comparative analyses or evaluations of EM methods.

Literature yields some work specifically dealing with approaches to analyzing and evaluating modeling methods. In [3, pp. 251-260] different techniques for evaluating modeling methods are classified and discussed. In [16, pp. 98-120], a classification of approaches to comparing (modeling) methods is proposed. More specifically, in [19], a particular framework for evaluating object-oriented modeling methods is presented. In [20], an evaluation technique is presented that builds on computing metrics for language specifications.

When it comes to specific examples of analysis, a comparative assessment of several EM methods is performed in [1, pp. 957-960]. The evaluation schema is based on a conception of EM set forth in the same paper [1, pp. 943-946]. The assessment includes several methods and criteria considered here as well,

but the main goal of the paper is the elaboration of a particular EM method. In [17], an in-depth comparison of three EM methods is offered, taking into account architectures, meta models, and process models. None of the methods considered here is assessed in this work. In [20, pp. 3403-3406], different modeling methods (the scope is not limited to EM) are compared based on ‘formal aspects’. The aim of this analysis does not correspond with the one of our paper, because it focuses on the degree of formalization with which syntax and semantics of modeling methods are specified.

With a focus on enterprise architecture (EA), an overview of the state of the art is provided in [21]. The overview also includes a comparative summary of different EA and EM methods [21, pp. 294-295], but it does not point out particularities of the underlying modeling languages. A comprehensive analysis of 22 different EA approaches is offered in [22]. Each approach is summarized and discussed in detail [22, pp. 21-112]. However, the approaches are not contrasted with respect to the offered modeling concepts. Less comprehensive high-level overviews of EA frameworks and methods are offered in, e.g., [23, pp. 11-41] and [24, pp. 65-73]. Neither one of the overviews provide a concrete juxtaposition of the methods. In addition to the mentioned overviews and comparisons, literature yields various contrasting discussions of specific pairs of methods (e.g., [25], [26]) while not including any further discussions of the state of the art in EM.

This paper distinguishes itself from the existing work with respect to three points. Firstly, it focuses on EM methods exclusively. It covers several current and distinct EM methods, which provide conceptual means for describing different aspects of organizational action systems and IS. This excludes more general (EA) frameworks. Secondly, the paper makes use of an analysis scheme that is explicitly defined in order to reach the defined goal. Thirdly, emphasis is placed on outlining and discussing conceptual and paradigmatic overlaps and distinctions between the considered EM methods. Thus, the paper contributes to harnessing complementary benefits in the further use, development, and integration of EM methods.

3 Analysis Framework

To conduct a comparative analysis of EM methods, we define and utilize an analysis framework (see Tab. 1) that accounts for the particularities of such methods. Considering the literature, we develop a framework in two steps. First, we relate our framework to the classifications suggested by [3, p. 251-260] and [16, pp. 111-113] in order to define its general orientation. In the next step, a particular analysis framework consisting of a set of categories (to delineate focal points of the analysis) and of particular criteria for each category is defined.

According to the classification proposed by [3, pp. 251-260], our analysis partly exhibits characteristics of a ‘feature comparison’ and a ‘theoretical and conceptual investigation’. An ‘empirical evaluation’ is not conducted. ‘Feature comparison’ describes an evaluation technique that makes use of predefined checklists of criteria [3, p. 251]. We use predefined criteria, but we do not regard them as

Table 1. Analysis framework

Criterion	Explanation
Way of thinking	
Background of the method	What was the initial motivation to develop the method? From which context did the method emerge?
Key goals and purposes	What are the key goals and purposes stated for the EM method?
Central assumptions	What are the fundamental assumptions of the EM method?
Way of modeling	
Concept specification	What are the mechanisms used to define modeling concepts?
Modeling concepts	Which central modeling concepts are defined?
Representation of modeling concepts	How can modeling concepts be represented (concrete syntax)? Are prototypical representation forms defined?
Way of working	
General approach to modeling	What is the general approach to applying the modeling method?
Process models and guidelines	Are any process models, procedural specifications, guidelines etc. available to guide users in applying the method?

checklist items. Instead, we use them to structure what is called ‘theoretical and conceptual investigations’ by [3, p. 251].

A classification scheme for analysis frameworks based on sets of criteria is proposed in [16, p. 98]. It can be used to explain the general orientation of our framework. Essentially, it is argued that analysis frameworks can be designed along two basic dimensions [16, pp. 102-105]. The first dimension relates to the criteria of a framework. The definition of criteria can vary with respect to their number and granularity [16, pp. 102-105]. The second dimension relates to the methods that are intended to be compared. The selection of methods can vary with respect to their number as well as the degree of their homogeneity [16, p. 105]. Based on these two dimensions, an analysis framework can be regarded as ‘horizontally dominant’ or ‘vertically dominant’ [16, pp. 111-113]. A ‘horizontally dominant’ framework takes into account a high number of criteria and methods [16, p. 111]. In contrast, a ‘vertically dominant’ one takes into account a low number of coarse criteria and homogeneous methods [16, p. 111]. The focus of a ‘vertically dominant’ comparison lies on identifying the essential characteristics of a set of methods [16, pp. 111]. As this focus is in line with our goal, the analysis framework devised below is aimed to be ‘vertically dominant’.

Building on this general orientation, we can particularize the analysis framework that will guide our comparison. The proposed framework consists of two levels: Criteria categories and concrete criteria. Table 1 indicates the considered categories and criteria. For each criterion, questions are provided that explain their intended scope. The criteria categories are adopted from a high-level view of IS development methods suggested in [27, pp. 13-25]. This view consists of the ‘way of thinking’, ‘way of modeling’, ‘way of working’, ‘way of controlling’, and ‘way of supporting’ [27, p. 13]. The categories defined by this view are also used to structure discussions and evaluations of modeling methods elsewhere (see, e.g., [24, pp. 76-78], [23, pp. 51-52]). In line with the goals of this paper, our focus lies on the ‘way of thinking’, ‘way of modeling’, and ‘way of working’. The ‘way of thinking’ perspective is related to the “*salient aspects* [...] of a

specific method” [27, p. 22]. It is supposed to embrace “starting points”, “basic objectives”, and “major assumptions underlying a certain approach” [27, p. 22]. The ‘way of modeling’ “structures the models which can be used in information systems development” [27, p. 14]. It thus pertains to modeling languages and modeling concepts defined by them. The ‘way of working’ corresponds with what is commonly referred to as the ‘process model’ or ‘procedure’ of a method (cf. [27, pp. 17-18]). ‘Way of controlling’ and ‘way of supporting’ are not within the scope of this paper, as the former is concerned with practical issues of the “management of a development process” [27, p. 20] and the latter relates to existing “collection of tools” [27, p. 24].

4 Comparative Analysis

4.1 Way of Thinking

Background. *ArchiMate*’s development started in 2002 within a project conducted by several Dutch companies, governmental organizations, and research groups [23, p. v], [4, p. xviii] with the aim to define a modeling language to be used in the context of EA management [23, p. x]. Its first version was described in 2004 [23, pp. ix-x], the most recent one in 2013 by The Open Group. Its design is oriented towards specifications of TOGAF [4, pp. 1-2, 14-15].

DEMO stems from an academic project started in the early 1990s [7,8,28]. It is motivated by the conviction that means for supporting business process design should focus on modeling their ‘essentials’, instead of the particular ways they are performed [7, p. 352]. Its design is inspired by language-related research strands in philosophy and other fields [28, pp. 237-247].

MEMO’s development started as academic research in 1989 and aimed at offering means for an integrative account of business-related and technological aspects to support the design, implementation, and use of IS [11, pp. 162-163]. It was motivated by the conviction that it is valuable to obtain different ‘perspectives’ on the business context of IS to be developed [11, pp. 163-169]. An object-oriented modeling framework was proposed [11, pp. 162-163], whose structure and focus has been adapted over time [12,1]. *MEMO* is progressively extended with domain-specific modeling languages (DSMLs) and methods (e.g., [29,30]).

WST is not explicitly labeled as an EM or EA approach, but it is based on similar considerations. It has been discussed and modified over the past two decades (e.g., [15,31,32,33]). *WST*’s development is driven by a dissatisfaction with (theoretical) conceptualizations of core terms in the IS field [31, pp. 1-3], [32, pp. 299-300], [15, p. 73]. *WST* regards a ‘work system’ as the central concept for considering possibly IT-supported systems in organizations [31, p. 7], [15, pp. 75-76]. Building on *WST*, the work system method (WSM) has been proposed to support the design and analysis of ‘work systems’ [15, pp. 83-84].

Key Goals and Purposes. *ArchiMate*’s goal is to define a modeling language for “the representation of enterprise architectures [...] as well as their motivation and rationale” [4, p. 2]. This reflects in the requirement that the

language should offer concepts for describing “main elements” of different organizational domains and relations between them [23, pp. 76-77]. It is stated that the language is aimed to be “as small as possible” [4, p. 4]. *ArchiMate* is not meant to replace detailed modeling languages for specific domains [23, p. 77]. As one key purpose, it is supposed to permit the creation of models, which are used as deliverables required by EA frameworks such as TOGAF [4, pp. 14-15].

DEMO is intended to support the ‘(re-)design’ and ‘(re-)engineering’ of organizational business processes [7, p. 360]. As a distinctive goal, it is stated that *DEMO* is aimed to enable the creation of an ‘essential model’ or ‘ontological model’ of organizational operations [7, p. 360], [8, p. 127]. Such a model is supposed to abstract “fully from the informational/documental as well as from the organizational (structural) realization” [7, p. 360]. The advantage of this approach is seen in that it enables to “look through the distracting and confusing appearance of an enterprise right into its deep kernel” [8, p. 12].

MEMO’s general goal is to bridge language-related gaps between stakeholders from business and IT-related domains [11, p. 162], [1, pp. 943, 945]. Initially, the focus was placed on supporting the development of IS that are “well integrated with a company’s strategy and its organization” [12, p. 73]. Currently, *MEMO* is also seen as a foundation for the construction of modeling methods in support of various domains [1, p. 950]. *MEMO* models are supposed to convey a non-deterministic view of action systems [1, p. 946]. At the same time, they are aimed to be transformable to implementation level code [1, p. 943]. *MEMO* aims to offer comprehensive tool support for model creation and use [12, p. 3], [1, p. 946].

WST: The key goal of *WST* and *WSM* is to support the design and analysis of what is considered, in a given situation, to be a ‘work system’ [15, pp. 83-84]. The application of *WSM* may be initiated by any kind of problem or opportunity noted for a ‘work system’, and may result in changes related to, e.g., business activities, participants, or software [15, p. 116].

Central Assumptions. *ArchiMate* implicitly follows assumptions linked to EA management, as it is proposed as a dedicated EA modeling language [4, pp. 1-2] [23, p. 75]. Explicitly, it is assumed that a set of generic concepts can be defined as generalizations of elements from all ‘layers’ of an enterprise (namely ‘business’, ‘application’, and ‘technology’) [23, pp. 78-79]. Also, it is assumed that ‘service’ is a suitable metaphor for describing “outputs” of elements from each layer, e.g., organizational units and applications [23, pp. 77-78]. Finally, it is assumed that limiting the number of concepts contributes to language usability, and that the current specification covers “most [EA] modeling tasks” [4, p. 4].

DEMO is stated to be based on the “single assumption” that “communication between human beings in organizations constitutes a necessary and sufficient basis for developing a theory of organizations” [34, p. 303]. This is detailed in a set of ‘axioms’ [8, pp. 81-125]. At first, it is assumed that operations of an organization are composed of ‘coordination’ (language-based interaction) and ‘production’ (realizing products or services) acts of human actors [8, pp. 80-88]. It is assumed that these activities can be assigned to one of three clearly distinguishable levels: ‘Forma’ acts relate to handling data; ‘informa’ acts relate to

interpreting information [8, pp. 106-114]. ‘Performa’ acts, on the coordination side, are stated to involve human commitments [8, pp. 106-114]. This view is inspired by, e.g., the speech act theory (Searle) and the theory of communicative action (Habermas) [7, pp. 352-354]. On the production side, the ‘performa’ level refers to the ability to “establish new original things” [8, p. 114]. An ‘ontological’ model of an organization, then, is defined to relate to the ‘performa’ level exclusively [8, pp. 106, 127]. To create such a model, it is consequently assumed that the state space and its possible transitions relating to relevant human ‘performa’ acts can be captured fully and formally [8, pp. 173-184]. Lastly, it is posited that starting from an ‘ontological’ model, implementation models regarding the operations and IS of an organization can be consistently ‘engineered’ [8, pp. 74-77].

MEMO’s main assumption is that organizational action and information systems should be analyzed and designed conjointly [1, p. 943], necessitating communication between different groups of organizational actors [1, p. 943], and that therefore integrated models are helpful that offer different ‘views’ or ‘perspectives’ on an enterprise [11, pp. 163 ff.], [12, p. 942]. It is assumed that the design of corresponding modeling languages can and should be based on detailed reconstructions of technical languages of the targeted actor groups [1, p. 945]. Similarly, it is assumed that (visual) notations being associative for targeted actors promote model acceptance and understandability [1, p. 943] [35, p. 55].

WST’s central assumption is that the concept ‘work system’ is suitable for thinking about, analyzing, and improving relevant sections of organizations [31, p. 7], [15, p. 75]. It is defined as a “a system in which human participants and/or machines perform work [...] using information, technology, and other resources to produce specific products/services for specific internal and/or external customers” [15, p. 83]. It is thus assumed that for the targeted problem cases in organizations meaningful definitions of concrete ‘work systems’ can be found.

4.2 Way of Modeling

Concept Specification. *ArchiMate*: Modeling concepts are introduced using several meta models [4, pp. 5, 17, 49, 63]. The underlying meta modeling language is not specified. The meta models do not specify all valid relationships between concepts [4, p. 18]. A complete definition of relationships is given in the form of additional tables [4, pp. 188-194]. Every relationship end features ‘0..*’ cardinalities [4, p. 5]. The specification of concepts does not include attributes. These are regarded as ‘profiles’ to be added on demand [4, pp. 137-138]. To explain concept semantics, textual descriptions are provided [4, pp. 18 ff.]. Extensions are specified using the same mechanisms [4, pp. 137-181]. It is noted that further extensions can be added on demand [4, pp. 9-10].

DEMO: The modeling concepts are defined using the Extended Backus-Naur Form (EBNF) and example diagrams [36, p. 2]. A complete meta model is not provided. The concept specifications do not include attributes. In [8, pp. 159-214], modeling concepts are introduced by means of an application scenario.

MEMO: Modeling concepts, their attributes, and relationships for each MEMO DSML are introduced using meta models presented in different publications. All meta models are defined using the MEMO meta modeling language (MML) [37]. Based on this common meta modeling language, relationships between concepts from different meta models are defined, leading to an integrated language architecture. As a result, concepts from all MEMO DSMLs can be used to create a comprehensive, integrated (enterprise) model.

WST does not define a modeling language, but uses glossaries to define concepts for describing nine core ‘elements’ related to work systems (e.g., [31, pp. 18-24], [15, pp. 79-81]). Additionally, textual ‘consistency rules’ for using the concepts are offered [15, p. 86]. Recently, a meta model has been published, defining and augmenting the core concepts. Attributes are not defined [26, p. 4].

Modeling Concepts. All investigated EM methods define specific sets of modeling concepts to address their stated goals. Due to the different concept specification mechanisms underlying each method, the semantics of the provided specifications are not fully comparable. Furthermore, concepts in different methods are conceptualized differently, even if they are denoted using the same name. Taking further into account that the total number of concepts is quite high (in particular, MEMO encompasses more than 150 concepts), a full comparison of all concepts and their details (i.e., attributes and relationships) is not in the scope of this paper. Instead, we provide in Table 2 a structured overview of (selected) modeling concepts of each method, which may be used to create models (thus, only non-abstract, i.e., instantiable, concepts are considered). The table has been constructed as follows. For ArchiMate, all 43 non-abstract concepts defined in [4] are listed, including those defined as part of two extensions [4, pp. 137-181]. Custom extensions are not considered, as they are not part of the official specification. For DEMO, the listed concepts have been derived from [36]. Because ENBF definitions and exemplary diagrams are used to define the language, some definitions can be interpreted as either concepts, attributes, or relationships. According to the best of our knowledge, and guided by the described key diagram types, we identified 17 conceptual definitions as core modeling concepts. The other definitions have been regarded as attributes, relationships, and further (syntactical) constraints. MEMO encompasses more than 150 concepts, therefore, only selected core concepts from recent MEMO DSMLs have been included in the table. References are provided that point to the full specifications. For WST, all 52 concepts defined in the most recent meta model [26, p. 4] are included. Concept names in Table 2 are taken literally from the specifications (names containing commas appear in quotation marks). The resulting list of concepts has been assigned to a coarse categorization of ‘areas’. The areas have resulted from interpreting and structuring the selected concepts. They serve purposes of clarity and abstraction solely, and are not proposed as a reference taxonomy. The area ‘Business aspects and resources’ is particularly broad, embracing traditional concepts from business administration (ArchiMate, MEMO, WST) as well as more general (‘ontological’) state and transition concepts used to describe (business) domains (DEMO) [36, p. 4]. Finally, it should be kept in

mind that even if a method does not feature a concept directly assigned to a certain area, this area could still be addressed implicitly. For example, highly general concepts such as ‘Fact Kind’ in DEMO can be used to model aspects related to various areas.

Representation. *ArchiMate* defines iconographic symbols to represent the concepts [4, pp. 46-47, 61, 75-76]. The symbols mostly consist of elementary

Table 2. Selected concepts of the analyzed EM methods

ArchiMate	DEMO	MEMO	WST
Area: Goals and strategies			
Value, Goal, Requirement, Driver, Assessment,		EngagementGoal, SymbolicGoal, GoalConfiguration (further concepts related to goals in [29]), Strategy [1]	Value Constellation, Value for Customer, Goal, Strategy, Enterprise Strategy, Department Strategy, Work System Strategy, Motive
Area: Business aspects and resources			
Business Object, Product, Contract, Business Service, Business Interface, Meaning, Representation, Constraint, Principle	Fact Kind, Derived Fact Kind, Product Kind, Scale	Product [29], Cost, Direct Cost Allocation, Proportional Cost Allocation, Unit-Based Cost Allocation, Cost Assessment, Agreement (further concepts related to costs in[30]), HumanResource, PhysicalResource, CompositeResource, TransportationResource (further concepts related to resources in [38])	Customer, Product/Service for Customer, Production/Service from Activity, Resource, Other Resource, Physical Entity, Time, Resource from Environment, Organizational Culture, ‘Laws, Standards, Regulations, Policies’, Other Env. Resource, Resource from Shared Infrastructure, Shared Human Resource, Shared Informational Resource, Shared Technical Resource
Area: Dynamic abstractions and human actions			
Business Process, Business Event	Transaction Kind, Aggregate Transaction Kind, Coordination Act, Production Act, Coordination Fact, Production Fact, Process Step, Discussion Step, Action Rule	BusinessProcess, ControlFlowSubProcess, Event, EventMerger, ProcessMerger, Exception, Branch, Branching, Fork, RegularSynch, ExclusiveSynch, MultiSynch (further concepts related to business processes in [39]), Decision, DecisionProcess, Stimulus (further concepts related to decisions in [40,41])	Business Process, Work System Activity
Area: Organizational structures, actors, and roles			
Business Function, Business Interaction, Business Collaboration, Location, Business Actor, Business Role, Stakeholder	Elementary Actor Role, Composite Actor Role, Scope of Interest, Responsibility Area	Organisation, OrganisationalUnit, Superior, Position, LocalUnitType, PrototypicalPosition, PositionShare, PositionCategory, Board, Committee, Role, Task, Interaction (further concepts related to organizational structures in [35]).	Work System (WS), Other Work System, Customer Work System, Enterprise, Participant, Actor Role, Skill/Capability, Knowledge/Expertise, Role in Customer Work System, Customer Participant, Non-Customer Participant
Area: Information systems and IT			
Application Service, Application Function, Application Interaction, Application Component, Application Collaboration, Application Interface, Application Object, Software System, Infrastructure Service, Infrastructure Interface, Infrastructure Function, Artifact, Node, Communication Path, Device, Network		Information System, IT Service, IT Functionality, IT Utilisation, IT Involvement, Software, Application Software, Database Management Software, Operating System Software, Cluster, Server, Personal Computer, Fax, Printer, Multi Device, Access Point, Router (further concepts related to IS and IT in [30])	Technological Entity, Tool, Transaction Record, Informational Entity, ‘Plan, Forecast, or Commitment’, ‘Guideline, Rule, or Structure’, Precondition, Trigger, Other Information, Document, Video, Image, Message, Conversation
Area: Risks, measurement, and indicators			
		Indicator, IndicatorCategory, Risk, Chance, Assignment, MeasureImpact (further related concepts in [42,43,30])	Performance Metric
Area: EA and EM processes and projects			
Plateau, Gap, Deliverable, Work Package			

graphical shapes augmented with small graphical metaphors (e.g., arrows). *ArchiMate* defines several so-called ‘viewpoints’, which describe how to select and arrange concepts to support certain ‘concerns’ [4, pp. 97-135].

DEMO models are represented using iconographical symbols, tables, and textual representations [36]. The graphical symbols consist of elementary geometrical shapes. For each kind of model, one or several representation forms (e.g., iconographical diagram types or table structures) are defined.

MEMO models are represented using iconographical elements, some of which are designed by a professional graphic artist [35, p. 55] [43]. The design of the notation is stated to be oriented towards recommendations proposed by [44]. For each DSML, at least one diagram type is specified [1, p. 947].

WST proposes a table structure (‘work system snapshot’) to represent descriptions of particular work systems [33, pp. 16-18], [15, p. 86]. No specific representation form for the specialized concepts in the meta models is defined.

4.3 Way of Working

General Approach. *ArchiMate* is to be applied as a part of more comprehensive enterprise architecture management processes [23, p. 75]. The general idea is to successively create and modify models of EAs (however conceptualized in particular settings) [23, pp. 4-6]. A corresponding prototypical ‘architecture life cycle’ should consist of the phases ‘design’ (encompassing the creation and analysis of models), ‘use’, ‘management’, and ‘idea’ [23, pp. 4-6].

DEMO: The general approach of applying DEMO is to develop an ‘ontological’ model of the business processes of an organization (or a part of it) and to use this model as the basis for various ongoing (re-)design and (re-)engineering efforts [8, pp. 74-77]. Since the aim of an ‘ontological’ model is to abstract from implementation and realization issues, it can be implemented in terms of different concrete activities or technological means [7, p. 362], [8, pp. 74-75]. Deriving more implementation-related models from an ‘ontological’ model is regarded as ‘engineering’ [8, p. 74]. Changing aspects of the ‘ontological’ model is regarded as ‘designing’. [7, p. 361]. ‘Engineering’, in contrast to ‘designing’, is understood as “not a matter of creativity but of craftsmanship” [8, p. 74].

MEMO: The general approach of MEMO is to create, maintain, and extend a comprehensive ‘multi-perspective’ enterprise model, and utilize this enterprise model for different purposes. Depending on the given needs, all or a subset of the existing MEMO modeling methods may be applied. It is assumed and suggested that a comprehensive enterprise modeling cannot be build ad hoc, but must be developed over time (e.g., [30, p. 382]). Once a (temporary version of an) enterprise model has been developed, it is proposed that a range of problems can be addressed [1, p. 950]. For example, it is suggested to perform goal planning processes [29], to support IT cost management [30], or to develop software systems from an existing enterprise model [45].

WST: The general approach of WSM is to investigate a particular work system, at a given point in time, to deal with identified problems or opportunities [32, pp. 21-22]. It is thus not the intent of WSM to develop and maintain more

comprehensive models over a longer period of time. As essential parts of analyzing a particular work system, it is proposed to create an ‘as-is’ and a ‘to-be’ snapshot of the work system under consideration [15, pp. 85-86, 114-116].

Process Models and Guidelines. *ArchiMate*’s specification does not include a process model or direct guidelines for applying the language. Instead, it is suggested that *ArchiMate* can be used in conjunction with the language-agnostic Architecture Development Method (ADM) defined by TOGAF [4, pp. 14-15]. Other publications contain some high-level guidelines and exemplary analysis techniques using *ArchiMate*, e.g., [23, pp. 115 ff., 189 ff.].

DEMO: A process model consisting of six steps is specified, which guides the general application of *DEMO* [8, pp. 144-158]. This process model is to be applied in an iterative manner. In addition to that, various general guidelines or and heuristics are available (e.g., [8, pp. 71-77]).

MEMO: For each *MEMO* modeling method, a specific process model is provided that guides the application of the language (e.g., [30, pp. 381-450]). In addition, various general guidelines are offered [11, pp. 337-342], [12, pp. 5-9]. Furthermore, a method guiding the development of new DSMLs is provided [43]. However, there is currently no overarching process model that guides the application of all existing *MEMO* methods in combination.

WST provides various specifications to guide the conduct of the WSM. The application of the method is described in detail in [33, pp. 21-32] and encompasses a process model consisting of several steps and elements, prototypical questions to be asked, and general guidelines. Further specifications include an overview of the suggested process model in the form of a business case template [15, p. 85] and a prototypical analysis schema [15, p. 116].

5 Discussion

Based on the preceding analysis, several similarities and distinctions between the considered methods can be identified.

Key Goals and Purposes. While the methods pursue similar goals on a basic level, their focal points vary. *DEMO* concentrates on a few elementary—‘ontological’—organizational aspects and is associated with the intention to ‘engineer’ an enterprise [8, p. 74]. This claim is not raised by other methods. Contrarily, *MEMO* emphasizes that an engineering point of view is not sufficient for shaping organizations, because, as is emphasized by organizational studies, organizations can be seen to emerge through social construction [1, p. 946]. *MEMO*, *ArchiMate*, and *WST* nevertheless intend to support organizational (re-)design efforts. *WST* is distinct in that its application is typically limited to a particular organizational (sub-)system [15, p. 83]. *ArchiMate* and *MEMO* both strive for creating comprehensive enterprise models. They differ, however, as *ArchiMate* is oriented towards use with existing EA frameworks and practices, whereas *MEMO* aims at supporting diverse organizational domains by providing dedicated DSMLs [1, p. 950]. In addition, as a distinct feature, *MEMO* models are aimed to be transformable into implementation level artifacts [1, p. 950].

Central Assumptions. The methods differ with respect to the degree that underlying assumptions are explicated and with respect to the research areas that serve as an inspiration or their basis. ArchiMate states only a few assumptions, mostly based on the EA literature [23, pp. 11-74]. DEMO refers to an explicit set of assumptions ('axioms' [8, pp. 81-125]), inspired by ontological philosophy and speech act theory [8, pp. 45-46, 83-85]. MEMO, in addition to fundamentals of (object-oriented) conceptual modeling [11, pp. 75-158], assumes the importance of taking into account several 'perspectives' on an enterprise and thus suggests considering insights and technical languages from different domains and research areas [1, pp. 944-945]. WST is based on assumptions building primarily on IS and business research [15, pp. 75-81]. In general, however, it appears that no method provides an exhaustive account of all underlying assumptions. To some degree, justifications, assumptions, and referenced theories are described in a partly selective manner, distributed over different publications. Moreover, when comparing all methods, we could not identify a common and coherent theoretical underpinning. For instance, while the importance of language is emphasized by DEMO and MEMO, it is not well elaborated upon which mechanisms are involved in the process of (linguistic) sense-making through EM, and the way that natural and modeling languages mutually influence and shape each other.

Concept Specification. The methods use different specification mechanisms to define modeling concepts. This is due to different backgrounds. Also, the meta modeling languages are not always made explicit, which threatens to impede the interpretation of formal semantics. Heterogeneous concept specification mechanisms limit the possibilities to analyze and integrate language specifications.

Modeling Concepts. The methods, firstly, vary with respect to the covered domains. Some domains are covered by all analyzed methods, e.g., operational business processes. DEMO, through the lens of language acts, even takes business processes as its prime focus. Here, emphasis lies on rigid routine operations in organizations. Other domains, such as indicators and risks (MEMO) or EA projects (ArchiMate), are covered only by some methods. In addition, some methods provide concepts that relate to less rigid, social or 'meaning-giving' facets of action in organizations (e.g., 'SymbolicGoal' in MEMO or 'Organizational Culture' in WST). Such concepts, however, do not seem to be widely adopted yet. Secondly, the methods vary with respect to the semantic richness of concepts. ArchiMate, DEMO, and WST favor a language design with fewer concepts and attributes, while MEMO intends to provide comprehensive reconstructions of the technical languages that prospective users are familiar with. This seems to point at an essential conflict regarding the design of modeling methods. On the one hand, a modeling language can be regarded as an analysis instrument, which suggests providing elaborate concepts that enable differentiated representations. On the other hand, a modeling language should be easy to use, which recommends using a small set of concepts that allow for a wider range of interpretations.

Representation. ArchiMate, DEMO, and MEMO all define iconographic symbols to graphically represent enterprise models. The symbols by ArchiMate and DEMO limit themselves mainly to basic shapes and figures, whereas MEMO aims

to offer visually richer symbols. DEMO, in addition to iconographic symbols and tables, also uses a textual representation. WST utilizes tables exclusively.

6 Conclusions and Future Research

In this paper, four EM methods have been analyzed using a configured framework. Based on the obtained results, we have pointed at their main similarities, distinctive features, and focal points. Obtained findings, on the one hand, contribute to the understanding of conceptual and foundational similarities and differences between the investigated methods. On the other hand, they allow us to suggest the following challenges that could be addressed in the future.

Comparative Analysis of Goals. Considering the variety of goals addressed by the investigated methods, an attempt could be undertaken to identify, structure, and compare goals that can possibly be addressed by EM methods, and means applied to achieve them. In particular, presumptions and world views underlying each goal could be pointed out, e.g., ‘engineering’ vs. ‘designing’ social systems.

Analysis and advancement of theoretical underpinning. Considering the heterogeneous assumptions and research strands associated with EM methods, it seems beneficial to clarify the theoretical underpinning of EM, and to enhance it with further insights from research areas such as the philosophy of language, organizational studies, and sociology. Particular attention could be devoted to the role and relation of natural languages and modeling languages to support bringing the design of EM methods yet closer to human (linguistic) perceptions of organizations. Furthermore, with some exceptions, the focus of current EM methods lies on more rigid aspects of organizing (i.e., operational processes and structures). It seems worthwhile to investigate which further insights from organizational studies (e.g., insights related to social aspects of organizing, or dealing with non-routine situations) could be considered in EM.

Common Meta Modeling Foundation. The analyzed methods utilize different concept specification mechanisms. To foster interpretation, comparability, and, consequently, a potential integration and extension of EM methods and research results, the usage of a common mechanism could be considered.

Language Expressiveness vs. Ease of Use. As there seems to be a conflict between method ease of use and support for differentiated analyses, an attempt could be undertaken to investigate approaches to mitigating this conflict.

Cognitive Perception. EM methods have different goals, target partly different user groups, and make use of different concrete syntaxes. This could suggest investigating the suitability of different graphical representation forms, especially generic vs. context-specific visualizations, for different purposes and user groups.

To gain a more thorough view of the current state of EM, however, a more comprehensive analysis with a more detailed analysis framework is necessary. Such an analysis remains on our research agenda.

References

1. Frank, U.: Multi-Perspective Enterprise Modeling: Foundational Concepts, Prospects and Future Research Challenges. *SoSyM* 13(3), 941–962 (2014)
2. Stirna, J., Persson, A.: Evolution of an Enterprise Modeling Method – Next Generation Improvements of EKD. In: Sandkuhl, K., Seigerroth, U., Stirna, J. (eds.) *PoEM 2012. LNBP*, vol. 134, pp. 1–15. Springer, Heidelberg (2012)
3. Siau, K., Rossi, M.: Evaluation techniques for systems analysis and design modelling methods – a review and comparative analysis. *ISJ* 21(3), 249–268 (2011)
4. The Open Group: *ArchiMate 2.0 specification: Open Group Standard*. Van Haren, Zaltbommel (2012)
5. Scheer, A.W.: *ARIS - Modellierungsmethoden, Metamodelle, Anwendungen*, 4th edn. Springer, Heidelberg (2001)
6. Hubert, Ö., Robert, W. (eds.): *Business Engineering. Auf dem Weg zum Unternehmen des Informationszeitalters*, 2nd edn. Springer, Berlin (2003)
7. Dietz, J.L.G.: Demo: Towards a discipline of organisation engineering. *EJOR* 128(2), 351–363 (2001)
8. Dietz, J.L.G.: *Enterprise Ontology: Theory and Methodology*. Springer, Berlin (2006)
9. Rolland, C., Nurcan, S., Grosz, G.: Enterprise knowledge development: the process view. *Information & Management* 36(3), 165–184 (1999)
10. Sandkuhl, K., Wißotzki, M., Stirna, J.: *Unternehmensmodellierung: Grundlagen, Methode und Praktiken*. Springer, Heidelberg (2013)
11. Frank, U.: *Multiperspektivische Unternehmensmodellierung: Theoretischer Hintergrund und Entwurf einer objektorientierten Entwicklungsumgebung*. Oldenbourg, München (1994)
12. Frank, U.: Multi-Perspective Enterprise Modeling (MEMO): Conceptual Framework and Modeling Languages. In: *Proceedings of the 35th HICSS* (2002)
13. Ferstl, O.K., Sinz, E.J.: Modeling of Business Systems Using the Semantic Object Model (SOM). In: Bernus, P., Mertins, K., Schmidt, G. (eds.) *Handbook on Architectures of Information Systems*, pp. 339–358. Springer, Berlin
14. The Open Group: *TOGAF Version 9.1*. Van Haren, Zaltbommel (2011)
15. Alter, S.: Work System Theory: Overview of Core Concepts, Extensions, and Challenges for the Future. *JAIS* 14(2), 72–121 (2013)
16. Strahringer, S.: *Metamodellierung als Instrument des Methodenvergleichs. Eine Evaluierung am Beispiel objektorientierter Analysemethoden*. Shaker, Aachen (1996)
17. Leist-Galanos, S.: *Methoden zur Unternehmensmodellierung. Vergleich, Anwendungen und Integrationspotentiale*. Logos, Berlin (2006)
18. Frank, U.: Essential Research Strategies in the Information Systems Discipline: Reflections on Formalisation, Contingency and the Social Construction of Reality. *The Systemist*, 98–113 (1998)
19. Frank, U.: Ein Bezugsrahmen zur Beurteilung objektorientierter Modellierungssprachen – veranschaulicht am Beispiel von OML und UML. *Technical Report 6*, Universität Koblenz-Landau, Koblenz (1997)
20. Bork, D., Fill, H.G.: Formal Aspects of Enterprise Modeling Methods: A Comparison Framework. In: *Proceedings of the 47th HICSS*, pp. 3400–3409 (2014)
21. Aier, S., Riege, C., Winter, R.: *Unternehmensarchitektur – Literaturüberblick und Stand der Praxis*. *Wirtschaftsinformatik* 50(4), 292–304 (2008)

22. Buckl, S., Schweda, C.M.: On the State-of-the-Art in Enterprise Architecture Management Literature (2011)
23. Lankhorst, M.: Enterprise Architecture at Work: Modelling, Communication and Analysis, 3rd edn. The Enterprise Engineering Series. Springer, Heidelberg (2013)
24. Land, M.O., Proper, E., Waage, M., Cloo, J., Steghuis, C.: Enterprise Architecture: Creating Value by Informed Governance. Springer, Berlin (2009)
25. Ettema, R., Dietz, J.L.G.: ArchiMate and DEMO – Mates to Date? In: Albani, A., Barjis, J., Dietz, J.L.G. (eds.) CIAO! 2009. LNBIP, vol. 34, pp. 172–186. Springer, Heidelberg (2009)
26. Alter, S.: Potentially Valuable Overlaps between Work System Theory, DEMO, and Enterprise Engineering. In: 1st Workshop on Enterprise Engineering Theories and Methods, IEEE Conference on Business Informatics 2014, Geneva, pp. 1–8 (2014)
27. Wijers, G.M.: Modelling Support in Information Systems Development. PhD thesis, Technische Universiteit Delft, Delft and Netherlands (1991)
28. Dietz, J.L.G., Widdershoven, G.A.M.: Speech acts or communicative action? In: Bannon, L., Robinson, M., Schmidt, K. (eds.) Proceedings of the 2nd ECSCW 1991, pp. 235–248. Kluwer, Dordrecht (1991)
29. Overbeek, S., Frank, U., Köhling, C.: A language for multi-perspective goal modelling: Challenges, requirements and solutions. CSI 38, 1–16 (2015)
30. Heise, D.: Unternehmensmodell-basiertes IT-Kostenmanagement als Bestandteil eines integrativen IT-Controllings. Logos, Berlin (2013)
31. Alter, S.: A general, yet useful theory of information systems. CAIS 1(13), 1–70 (1999)
32. Alter, S.: Work Systems and IT Artifacts - Does the Definition Matter? CAIS 17(14), 299–313 (2006)
33. Alter, S.: The Work System Method: Connecting People, Processes, and IT for Business Results. Work System Press, Larkspur (2006)
34. Dietz, J.L.G.: The atoms, molecules and fibers of organizations. Data & Knowledge Engineering 47(3), 301–325 (2003)
35. Frank, U.: MEMO Organisation Modelling Language (1): Focus on Organisational Structure. ICB-Research Report 48, University of Duisburg-Essen (2011)
36. Dietz, J.L.G.: Demo-3: Models and representations, version 3.7 (2014)
37. Frank, U.: The MEMO Meta Modelling Language (MML) and Language Architecture. 2nd Edition. ICB-Research Report 43, University of Duisburg-Essen (2011)
38. Jung, J.: Entwurf einer Sprache für die Modellierung von Ressourcen im Kontext der Geschäftsprozessmodellierung. Logos, Berlin (2007)
39. Frank, U.: MEMO Organisation Modelling Language (2): Focus on Business Processes. ICB Research Report 49, University of Duisburg-Essen (2011)
40. Bock, A., Kattenstroth, H., Overbeek, S.: Towards a modeling method for supporting the management of organizational decision processes. In: Proceedings of the Modellierung 2014. LNI, vol. 225, pp. 49–64. Gesellschaft für Informatik, Bonn (2014)
41. Heß, M., Schlieter, H., Täger, G.: Modellierung komplexer Entscheidungssituationen in Prozessmodellen – Anwendung am Beispiel der Tumorklassifikation bei Weichteilsarkomen. In: Thomas, O., Nüttgens, M. (eds.) Dienstleistungsmodellierung 2012, pp. 268–290. Springer, Wiesbaden (2012)
42. Strecker, S., Heise, D., Frank, U.: RiskM: A multi-perspective modeling method for IT risk assessment. ISF 13(4), 595–611 (2011)

43. Frank, U.: Domain-specific modeling languages: Requirements analysis and design guidelines. In: Reinhartz-Berger, I., Sturm, A., Clark, T., Cohen, S., Bettin, J. (eds.) *Domain Engineering*, pp. 133–157. Springer, Berlin (2013)
44. Moody, D.L.: The Physics of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering. *IEEE TSE* 35(6), 756–779 (2009)
45. Gulden, J.: *Methodical Support for Model-Driven Software Engineering with Enterprise Models*. Logos, Berlin (2013)