Digital Transformation Designer: Towards a Comprehensive, Collaborative and Easy-to-Use Modeling Support for Enterprise-level Change Endeavors

Tobias Kautz[*,a], Robert Winter[a]

[a] University of St. Gallen, Institute of Information Management, St. Gallen, Switzerland

Abstract. *Established companies intending to leverage digital technologies are required to innovate their 'legacy' business models through organizational transformations. Existing modeling support often leaves a 'white space' between informal canvas-style models used in the early phases and (semi-)formal aspect models used in the later phases of transformation endeavors. Adopting a Design Science Research approach, this paper presents a semi-formal model that is intended to fill this gap, i. e., to provide easy-to-use support for the heterogeneous stakeholders that participate in early phases of digital transformation endeavors. Being based on the traditional Business Engineering set of models and methods, this comprehensive and collaborative approach was validated together with the digital transformation program manager of a large, international corporation. For supporting analysis, reflection and design tasks that involve a broad range from canvas-style models to enterprise architecture models, four requirements were identified to be central: (remote) collaboration support, a holistic and integrative perspective, an enterprise-level view, and a focus on change. The actual model is comprised of over 20 partial models including popular canvases depicting the transformation program's content on the strategy-to-IT layers, the enterprise and local level, and in the as-is and to-be state. Demonstration and evaluation were done with practitioners and students of an Executive Master program focusing on digital transformation. Both confirmed the utility of the underlying method and recognized its distinctive features, while capability and IT landscape models were found especially relevant. The method is expected to be applicable for digital transformations beyond the case and also to be projectable to smaller-scaled digital business innovations.*

Keywords. Digital Transformation • Program Management • Enterprise Modeling • Business Engineering • Enterprise Architecture

Communicated by Henderik A. Proper, Giancarlo Guizzardi. Received 2022-07-01. Accepted on 2022-12-10.

1 Introduction

In order to leverage digital technologies, digital transformation (DT) has become an important need also for established, successful companies – but is also particularly challenging for them. DT can be defined as a "profound "..." transformation of business activities "...", competencies, and models to fully leverage the changes and opportunities brought by digital technologies "..." in a strategic and prioritized way" (Demirkan et al. 2016, p. 14). This implies that the change is a more fundamental one than the one required by mere digitalization and concerns the business logic, value creation, ecosystem interactions, and core competencies of a firm (Verhoef et al. 2021, pp. 891–892). A DT can be implemented in various forms, each

* Corresponding author.
E-mail. tobias.kautz@unisg.ch
We thank the participants of the "Models-at-Work" session during the 15th IFIP WG 8.1 Working Conference on the Practice of Enterprise Modelling (PoEM 2022) and the anonymous reviewers for their valuable feedback.

with a different focus, that are (non-conclusively) distinguished by Jöhnk et al. (2022, p. 3) as follows: innovation-focused like digital accelerators, delivery-focused such as digital units, or change-focused like cross-departmental programs. As we expect that this has only limited impact on the considerations made in this paper, we use the term DT 'endeavor' whenever referring to the general notion.

MedComp (pseudonymized for confidentiality reasons), a large company (> 10'000 employees worldwide) which manufactures 'small' items like surgical equipment or implants, but also 'large' medical machinery (e. g., for diagnostics) was facing exactly the situation described above. Traditional competitors, which were inferior in mechanical quality of the products but starting to provide digital solutions, set MedComp under pressure. At the same time, new, 'digital-born' companies and / or companies with huge investment budgets entered the market. Patients were empowered through digitally enabled self-diagnostics opportunities as well as regulations that emphasize a health intervention's outcome. Thus, MedComp decided to embark on what it called its *Digital Acceleration Journey* with a dedicated program. This way, the already existing, numerous individual digitalization use cases should be approached in a systematic manner along three *Lighthouses* focused on digital services for B2C and B2B customers as well as supply chain optimization, enabled by a strong technology backbone. After defining a *Digital Ambition* and conducting a *Digital Readiness* assessment, the company developed a *Playbook* to capture the future state design of the organization. However, MedComp relied on text protocols of presentation decks and has not chosen to use more formal, more coherent representation of their analyses and plans.

This might not only be the case for MedComp, but indeed for many companies. When thinking about important tools, i. e., "any kind of object, concept, framework, method, or model that helps practitioners analyze and solve problems, make decisions, and collaborate with others" (Avdiji et al. 2020, p. 696) for the management of such DT endeavors, one might not think of (semi-formal) modeling. Instead, one could think of decision making and strategic planning techniques, e. g., scenario analysis, or, closer to implementation, e. g., of software tools for portfolio and project management. At best, one might think about canvases, which in academia are also called "visual inquiry tools "..." [that] provide a shared and framed design space in which practitioners can jointly inquire into a strategic management problem" (Avdiji et al. 2020, p. 696). In the context defined above, for most stakeholders 'classical' (i. e., more or less formal) modeling becomes relevant late in the process, when business concepts are to be translated or aligned with software support, for instance by modeling processes using semi-formal BPMN or UML diagrams to describe the behavior of a software application.

Formal modeling languages can be understood as the ones providing a conclusive set of symbols with a clearly-defined syntax and semantics, whereas *semi-formal* modeling languages are characterized by a conclusively defined set of symbols with an at least partially clearly defined syntax and a rudimentary, e. g., natural language specified semantics (Frank and van Laak 2003, pp. 20–21; Vogel et al. 2009, p. 267). When thinking of an *informal* modeling language as one which only provides some symbols whose syntax and semantics depend on the ambiguous and subjective interpretation of the model users (Frank and van Laak 2003, p. 21), canvases can be considered to be informal models as well. More concrete, canvases provide a set of symbols (i. e., sticky notes), but *like they are often applied*, they give limited guidance on which relations between the sticky notes (and their respective entity types) are allowed, nor what the sticky notes and their relationships imply semantically – especially, when different canvases are used together. This interpretation is then dependent on the varying experience and skills of the canvas user.

Even by the above given definitions of semi-formal and informal modeling which we have chosen as they explicitly link to the constituting elements of a modeling language, it becomes clear that these two approaches are not naturally compatible and could create challenges when respective models are directly linked together (e. g., to support directly related analysis steps).

The question arises how this conceptual gap (the 'white space' [1] between different modeling paradigms used in a DT endeavor) can be bridged, thereby avoiding incompatibilities and providing seamless support for 'business-to-IT' analysis and design needs – which are at the core of DT endeavors. Thus, we first need to understand the problem class MedComp was facing.

2 Problem understanding

In general, DT endeavors can be analyzed with different theoretical and inquiry lenses, each with a different focus. For example, with an emphasis on why to transform, how DT endeavors evolve in terms of a process model, or their success factors (Verhoef et al. 2021, p. 890). However, in order to develop a concrete model to be used by practitioners and the general approach behind it, a less abstract view allowing to create pragmatic, 'how to knowledge' (Baskerville and Pries-Heje 2010, p. 272) is needed. That is why our research perspective understands DT as a set of practices, which can be defined as "recurrent structured activities that people perform to get their work done" (Levina and Orlikowski 2009, p. 673). Among other properties, practices are characterized by their purpose-orientation, their collective, normative, social, and material nature, their adaptivity to changing circumstances, and the involvement of power between actors (Nicolini and Monteiro 2016, pp. 111–114). These decision-making and change management practices can be made tangible in terms of use cases that are defined through specific stakeholders collaborating to achieve a beneficial outcome for the DT endeavor. Depending on this outcome, as well as the concrete procedure and the involved stakeholders (cf. the constituting elements of a practice), each use case poses requirements for 'functional and non-functional' properties of the model. The functional properties can be abstracted to generic purposes of the model, which are needed and fulfilled differently strong depending on the concrete use case. A rough overview of these use cases is shown in Tab. 1.

However, based on the above given definition of a practice, these use cases also share several commonalities: They are *collaborative*, i. e., "involving, or done by, several people or groups of people working together" (Oxford Advanced Learner's Dictionary 2022) that most probably have heterogeneous educational and professional backgrounds. Therefore, supporting tools should be simple enough to be quickly comprehended by these heterogeneous stakeholders. Second, they require a *holistic and integrative perspective*, i. e., to consider multiple entities along the strategy, organization, alignment, and IT layer and especially their relationships (cf. the definition of DT given in se.1). Use cases in the context of a DT also need the breadth of an *enterprise-level view*, i. e., to consider either the whole organization or independent units as a whole, or their parts (e. g., a business unit, product, endeavor) but in relation to this whole. In concrete, this implies consistency with entities of the respective types already existing in the organization and their reuse where appropriate. Last, given the nature of (digital) transformations, they call for a *change perspective* looking at the as-is and to-be situations highlighting gaps to be closed.

Canvases such as the Business Model Canvas (BMC) (Osterwalder and Pigneur 2010, p. 44) are collaborative by nature. This should not mean that people working with a canvas for the first time are directly able to create 'high-quality' results. However, the notation of a canvas is less complex than the one of most semi-formal models, and canvases use dedicated elements (e. g., metaphors, icons, or trigger questions) that facilitate understanding and application (Avdiji et al. 2020, p. 718). This also the reason why 'just' relaxing some syntactical

[1] Please note that we use this term in analogy to Rummler and Brache (2012, p. 18) referring to a missing integration between organization-internal activities (often executed by different functions) that reduces the efficiency and effectiveness of output creation.

constraints for semi-formal models at the beginning before iteratively refining them will not per se make them easier to be applied collaboratively. In terms of the BMC, they could also have a very high-level holistic perspective, e. g., looking at the value proposition, value chain, and value capturing. But they do not explicitly highlight the relationships between their elements, given their normally paper-made, 'sticky-based' nature. And as they are usually applied in settings of ideation, i. e., "a process by which new ideas are elicited and discussed "..." identifying multiple alternative solutions for a task, problem, or concern being pursued" (Maaravi et al. 2021, p. 1433), they are rather focused on a vision of the future state. In these ideations, generating lots of novel ideas is usually also favored over consistency and re-use required for the enterprise-level view. In contrast, formal models and 'classical' semi-formal models (like the ones created using BPMN) can cover the as-is and to-be perspectives as well as promote consistency and re-use, often facilitated by software tools. In case of holisticness, they often concentrate on one of the abovementioned layers, while exceptions might exist in the Enterprise Architecture (EA) domain. However, while some of them – at least in theory – could be applied collaboratively, all collaborators need specialized knowledge – which cannot (and should not) be assumed in the context of DT. To conclude, there is a need for an improved approach (i. e., a modeling method) that bridges the white space and can be used to holistically describe, analyze, and support the design of complex transformations across the strategy, organization, alignment, and IT layer.

We understand a method as "an approach to perform "..." systems development "...", based on a specific way of thinking, consisting of directions and rules, structured in a systematic way" (Brinkkemper 1996, pp. 275–276). More concretely, a method consists of (Gutzwiller 1994, pp. 13–14) development / design *activities*, performed by a specific *role*, that create or use *results* (in this case, especially models). These results are linked through a *meta-model*. A *technique* is a set of instructions that support creating results. Usually, techniques focus on certain aspects of design activities and thus can be re-used for different activities. A specific class of techniques are modeling techniques, which comprise a *modeling language* and a *modeling procedure*, where the first can be further subdivided in a *notation*, a *syntax*, and *semantics* (Karagiannis and Kühn 2002, p. 3).

Given the amount of resources invested in DT endeavors as well as the range of internal and external stakeholders affected if these endeavors fail, investigating how tools can support decision-making practices in this context, is a relevant problem. Although these tools are intended to be used by a variety of stakeholders, the focal stakeholders of the problem are top-level executives and DT managers (e. g., program managers) aiming at reaching the DT's overall objectives. One means to achieve this is providing a methodological framework (i. e., suitable tools) for the DT endeavor. The problem's significance is intensified through the recent DTs initiated or accelerated by Covid-19 and the ongoing need for enterprise-level changes in the context of what is called organizational agility, even if they are not realized in the setting of a full-blown DT endeavor (Teece et al. 2016, p. 13). Concretely, enterprises might conduct *digital business innovation* endeavors that still affect all of the layers mentioned earlier in this sect.(i. e., same 'depth' as DT endeavors) but have a smaller scale such as a specific product (i. e., narrower 'breadth'). Please refer to Kautz and Winter (in press) for a more detailed discussion of this aspect, also in conjunction with the form of implementation for the DT endeavor (see Sect. 1).

To gain first insights and experiences how a method filling the 'white space' explained above might look like, we developed a holistic, semi-formal model serving as an reflection tool for the program manager of MedComp's DT program, following the Design Research Approach by Peffers et al. (2007). This means, that our approach to develop the model (cf. sect.4) is a good starting point for a generally applicable modeling method for DT endeavors. Based on the evaluation of the first design iterations, future research needs to abstract from case specifics and integrate additional design support aspects (see sect.7.2).

As secondary purposes, the model was also (i) used as a support artefact in an executive education program for DT, and it (ii) served as a case for scientific inquiry. While the direct efforts to create the presented model took place recently (cf. sect.4.2), it draws on the insights from two decades of research and executive teaching in the context of Business Engineering (BE). Thus, this report will switch (where appropriate) between the general level (teaching and research) and the instance level (situated practical use). This facilitates a more fundamental understanding and highlighting general research issues and opportunities.

The generic purposes on which the use cases built are based on Leist (2002, pp. 8–9) and can be specified as follows:

- *Problem understanding / sensemaking* means supporting already knowledgeable stakeholders to deepen their understanding by means of a representation that is more complete, better relates to their concerns, etc.
- *Problem analysis* refers to the in-depth investigation of a certain aspect, e. g., to identify potentials for improvement or innovation.
- *Planning, steering & controlling of activities* indicates that the model should be used as a foundation for aligning complementary and derived models (e. g., for key performance indicators).
- *Design (of problem solution)* indicates that the model should facilitate the development of new organizational entities such as processes or information systems.
- *Communication support (common language)* implies that the model should provide a common syntax and semantic to enable stakeholders with different backgrounds and knowledge asymmetries to discuss a certain matter without misunderstandings and on a common level of abstraction.
- *Training* concerns the usage of the model to highlight the to-be state (both prescriptive, e. g., for newcomers, and after changes occurred) of the organization or information system to outsiders (i. e., not working in the domain were the model originated) to highlight the implications on their concrete way of working and enable the usage of a system (if applicable).

The case for which this article's model was developed constitutes one of these use cases. The use case *reflection tool for DT managers* comprises the qualitative assessment and identification of improvement potentials of previously made design choices with regards to the business-to-IT-solution (i. e., the business and corresponding IT architecture) which is implemented by the endeavor, often in collaboration with peers. This way, DT managers can, for example, check whether missing architectural elements are addressed through corresponding development projects, and relationships between the elements are reflected in the collaboration between the projects. Therefore, it primarily combines the generic purposes of *problem understanding / sensemaking* and *problem analysis*, where the problem in this context lays in appropriately structuring the implementation activities and finding general improvement potentials from an implementor's perspective. Regarding the first purpose, the model provides an alternative, reduced and fundamental view on the endeavor's actual business content as opposed to the presentation slides and associated documents that DT managers are used to. Regarding the latter, the effort to develop this alternative model and perspective is justified by the aim to generate insights regarding potentially missing elements and unaddressed relationships that would have not been gained without this re-conceptualization.

Building on that, a secondary purpose of developing the model was to support DT researchers to understand how such types of models can be used to, e. g., easier and / or more comprehensively, understand, challenge and design DT endeavors. As this constitutes a meta-perspective, the generic purposes do not apply to this. Last, it should also

Use case	Primary purposes					
	Problem understanding / sensemaking	Problem analysis	Planning, steering & controlling of activities	Design (of problem solution)	Communication support (common language)	Training
Systematic identification of improvement opportunities (process, product / service, business model)	x	x		x		
Design of business-to-IT-solutions				x	x	
Reflection of previously designed business-to-IT-solutions	x	x				
Endeavor planning incl. its communication	x		x			
Performance management for endeavors			x		x	
Creation of business / IT alignment	x				x	
Trainings in context of change management					x	x
Architectural governance			x			x

Table 1: Exemplary use cases in DT potentially requiring modeling support. ***Note:*** *The primary purposes are based on Leist (2002, pp. 8–9).*

be used to convey the BE approach for DT and digitalization endeavors to students of an Executive Master program specialized on digital business innovation, also leveraging the insights gathered from the other two purposes. This corresponds most likely to the generic purpose of *communication*, given the heterogeneous background of students.

In the following, we elaborate on how the model was developed and which research conclusions could be made on this basis. Thus, the remainder of the paper is structured as follows: After briefly reviewing the existing literature in sect.3, the general methodology as well as the evidence and activities / efforts used to create the model are outlined (sect.4), and in sect.5 the model is presented in terms of concrete requirements and the resulting modeling method as well as its output (i. e., the model itself). Then, we present the results of the different demonstrations and evaluations it underwent (sect.6). The article ends with a discussion and conclusion in sect.7, commenting on the Return on Modeling Effort (RoME) and the reflection both for the research team and with regards to further research opportunities.

3 Related work

To validate the supposed white space between lightweight design support tools and complex aspect models, a systematic literature search was conducted. The databases *Web of Science*®, the *AIS eLibrary* (AISeL), and *Scopus*® (the latter limited to the *LNBIP* and *LNISO* book series) were used to search for papers which have the following keywords in the title, abstract, or keywords[2] :

- *AB=(digital transformation OR digitalization OR enterprise transformation OR business transformation OR organi?ational transformation OR business innovation OR business*

[2] The shown search strings are exemplary ones for the Web of Science® and shown for abstracts only. All search strings can be found under: https://doi.org/10.5281/zenodo.7318057

model innovation OR digital innovation OR organi?ational change OR digital business) AND AB=(model$ing NEAR/0 (enterprise OR conceptual OR domain))

- *AB=(strateg*) AND AB=(model$ing NEAR/0 (enterprise OR conceptual OR domain)) NOT AB=(digital transformation OR digitalization OR enterprise transformation OR business transformation OR organi?ational transformation OR business innovation OR business model innovation OR digital innovation OR organi?ational change OR digital business)*
- *AB=(participat* OR collaborative OR interactive OR natural OR light$weight OR grass$root*) AND AB=(model$ing NEAR/0 (enterprise OR conceptual OR domain)) and Computer Science Information Systems or Computer Science Theory Methods or Computer Science Software Engineering or Computer Science Interdisciplinary Applications or Operations Research Management Science or Management or Information Science Library Science or Business or Social Sciences Interdisciplinary or Economics or Psychology Multidisciplinary or Ergonomics or Business Finance (Web of Science Categories)*

Moreover, we searched for papers citing Sandkuhl et al. (2018) as this is an important research agenda for more lightweight modeling formulated by several members of the enterprise modeling community.

Aier and Gleichauf (2010, pp. 59, 65) provide a method for deriving a transformation endeavor structure based on existing as-is and to-be EA models and thus explicitly do not consider how to develop those. However, this seems to be a very valuable approach for the endeavor planning use case mentioned in Tab. 1, which is not extensively covered in this paper. de Kinderen et al. (2021, pp. 2–3, 27) propose a modeling method integrating strategic, goal- and IT-related aspects, which is, however, intended to be used for the ex-ante or ex-post analysis of DT strategies – replacing SWOT – and not for the (more detailed) design of business-to-IT-solutions. Hafsi and Assar (2021, pp. 6–8) provide the *DT Oriented Model Canvas*, which provides external-influences-, process-, infrastructure-, capability-related aspects in one overview that should be suitable for executives. However, this seems to be more useful for the summary of an existing design instead for its development.

There were also highly specialized approaches found like the ones by Bork et al. (2016) for smart cities or by van Gils and Weigand (2020) for sustainable DT that should not be elaborated here. Fill (2020, pp. 1, 3) sketches an approach to model the impact of technological developments in a post-DT state called “digital ubiquity” (p. 1), which is a different context than the one of this research.

Going more into the individual constituting elements of a modeling method, van Gils and Proper (2018, pp. 260–263, 267–270) derive challenges and recommendations for EA modeling languages used in DTs. While our research does not (yet) intend to develop a fully specified modeling language, issues identified by them such as value co-creation, an appropriate level of model granularity, lean and business-user-oriented modeling languages, consistent layering, and capturing different nuances of change might still be relevant. (McGinnis 2007, p. 141) considers enterprise modeling languages in general and highlights similar issues that could also be relevant for this research: next to different levels of model granularity and a business-user and use-case oriented modeling language he proposes to expand the scope beyond software-related but to human aspects and also to incorporate uncertainty perspectives.

Within the related domain of EA, Buckl et al. (2011, p. 5) derive three principles on how to capture change information in EA information models (regarding entities’ relationships to projects, their replacements, and lifecycle), which again only relates to the modeling language. Babar and Yu (2019, p. 7) derive requirements for enterprise modeling in the case of EA from the characteristics of DT. While most of them ultimately relate to the modeling language or to more specialized activities and results (e. g., different forms of scenario analyses), there are some fundamental requirements which might be interesting for the work in this paper: relating business and software processes, aligning enterprise and operational objectives, and incorporating the perspectives of customers and employees.

There is also research on how to develop more lightweight modeling languages. Bork and Alter (2018, pp. 122–124) provide general principles that balance the quality criteria of 'traditional' enterprise modeling with user-oriented ones. van den Oever et al. (2022, p. 3) propose a method on how to develop and integrate user-oriented enterprise models. Bider et al. (2021b) show how a modeling language can be adapted during application via the means of "modeling languages jargons" (p. 6) to cater the needs of specific user groups.

Tab. 2 provides an overview about selected (i. e., relevant) results of the related work just presented. Moreover, existing modeling approaches in the area of DT as well as the related fields of strategy and business innovation are shown, which were not mentioned above. For all sources, it is indicated to which extent they cover the common characteristics of use cases in a DT context described in sect.2. For the modeling approaches, coverage is assessed based on the meta-models and / or shown model instances in the respective source. Suitability for *collaboration* was proxied through the complexity (i. e., number of entity and relationship types), *holisticness* through having diverse entity types on the respective layers (Winter 2011, p. 27), the *enterprise-level view* through the possibility to (in principle) differentiate between the enterprise and business-to-IT-solution level, and the *change perspective* through either highlighting changes graphically and / or providing change-related entity types. To conclude, to our best knowledge there seems to be no already existing approach that fits to all common characteristics equally well, while there are plenty of requirements and solution components as well as similar solution approaches Drechsler and Hevner (2018, p. 89) already available.

4 Research method

4.1 In general

Faced with the need for pragmatic, how to knowledge (cf. sect.2), the research's nature of building something, i. e., an artifact, and the chance for a rigid evaluation, we chose a DSR approach. More precisely, the artifact in terms of the developed model corresponds to a result of an instantiated (modeling) method based on the general BE framework. This distinction also helps to understand the situated-general-switch made in this paper. While the research conducted to develop the modeling method can – overall and aggregated – be conceived as conducting relevance, rigor and design cycles (Hevner 2007, p. 88) the procedure to arrive at the situated model can be better captured by the DSR process model proposed by Peffers et al. (2007, pp. 52–56) depicted in fig.1.

4.2 Activities and effort

As a starting point, the problem was explicated as outlined above. The objectives were derived in qualitative terms from the problem space with regards to both the overarching purpose (i. e., effectiveness) and secondary traits (i. e., efficiency in the broader sense) (Alismail and Zhang 2017, p. 222; Peffers et al. 2007, p. 55). Special consideration was given to criteria for model quality (Becker et al. 2012, p. 32).

The design and development of the actual model involved the program manager of the case company (in the following, 'the practitioner') as well as a senior researcher / professor and a research associate / PhD student (both just summarized as 'the researcher', for simplicity). The activities comprised an initial semi-structured interview with the practitioner, the analysis of case-related documents, the selection of model types ('templates') and the modeling software, several iterations of model creation, and validation sessions with the practitioner. It should be noted, however, that

Name / description	Authors (exemplary)	Coverage of common use case characteristics							
		Collaborative	**Holistic & integrative**					**Enterprise-level view**	**Change perspective**
			Strategy	Organization	Alignment	IT	*Meta*		
Existing solution design knowledge (exemplary): general requirements (GR) and components (GC) and methods (M) for developing models in the DT context or lightweight models									
Distinguishing characteristics of enterprise modeling tools for enterprise transformation	McGinnis (2007)	GR					GR		GR
Challenges for enterprise modeling languages in the context of DT	van Gils and Proper (2018)	GR, GC					GR		GR, GC
Requirements for an enterprise modeling framework based on DT characteristics	Babar and Yu (2019)		*not applicable*				GR	GR	GR
Principles for relaxed enterprise modeling	Bork and Alter (2018)	GR					GR		
Modeling language jargons	Bider et al. (2021b)	GC					GC		
User-oriented enterprise models	van den Oever et al. (2022)	M					M		
Existing solution design entities: similar modeling methods in the area of strategy, business innovation, and DT; x = covered, (x) = partially covered									
Fractal Enterprise Modeling	Bider et al. (2021a); Lodhi and Bider (2019)	x	(x)	(x)		(x)		x	x
Modeling method for facilitating strategic analysis	de Kinderen et al. (2021)		(x)	(x)		x		x	x
KYKLOS Method	Koutsopoulos et al. (2022)	(x)	(x)	x		(x)		x	x
LiteStrat Modelling Method	Pastor et al. (2022)	x	x					x	
4EM	Sandkuhl et al. (2014)		x	x	x	x	*not applicable*	x	(x)
Modeling method for supporting strategic planning	Bergmann and Strecker (2018)	(x)	x	x				x	x
Work System Modeling Method	Bork and Alter (2020)		(x)	x		x		x	(x)
DT Oriented Model Canvas	Hafsi and Assar (2021)	x	(x)	x	x	x		x	x
Process Goal Alignment Technique	Roelens (2022)	x	x	(x)				x	x

Table 2: Overview of existing research. ***Note:*** *The differentiation into solution design knowledge and entities and their sub-categories is based on Drechsler and Hevner (2018, p. 89), while on the level of the sub-categories we prefer the terms general requirements and general components from Baskerville and Pries-Heje (2010, p. 274) over (meta-)requirements and (meta-)features used by Drechsler and Hevner (2018, p. 89). The layers shown in the 'Holistic & integrative' column originate from Winter (2011, p. 27).*

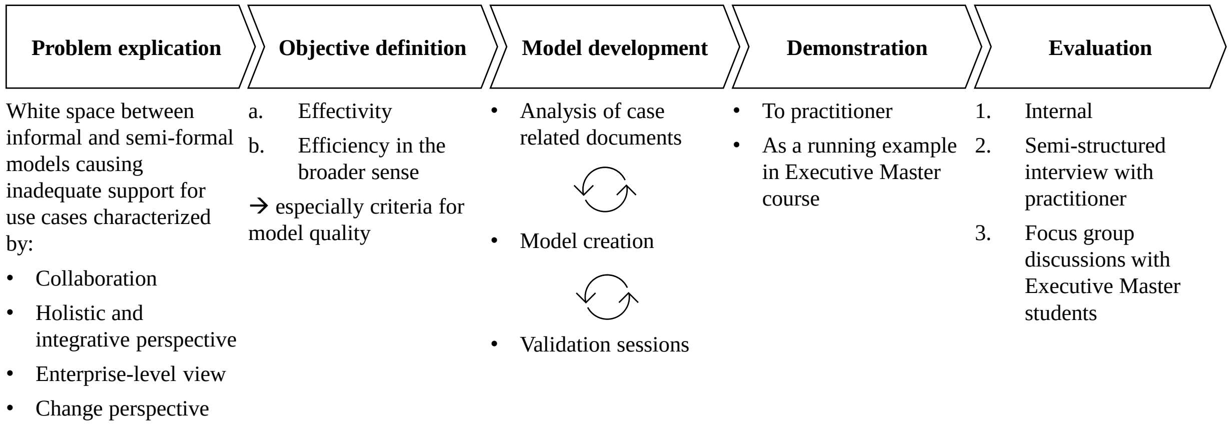

Figure 1: DSR approach deployed in this research. ***Note:*** *The approach is oriented at Peffers et al. (2007, pp. 52–56).*

this straightforward development was only possible on the basis of the comprehensive research conducted over two decades, which especially facilitated model type selection and modeling language development and as well provided applicable procedure models. A major achievement of preceding research was the BE framework's consolidation as a situational method documented as *Business Engineering Navigator* (Winter 2011), as well as several related master and PhD theses. It should be mentioned that the practitioner, as a graduate of the above-mentioned Executive Master program, was also working on this foundation. Given that the Business Engineering Navigator is more concerned with providing support for the latter, more sphases of DT endeavors, a more lightweight approach (cf. McGinnis 2007, p. 141; van Gils and Proper 2018, pp. 261–263) was developed using popular canvases. Several subsets of these canvases were applied in Executive Master endeavor's case projects – but never comprehensively and with focus on overall coherency, which was then done with developing this model.

The actual modeling activities required roughly 80 hours for the researcher and 6 hours for the practitioner. The evidence / sources used to develop the model consisted of the abovementioned interview data with the practitioner and company documents of the DT program, as well as supplementary public company information. The *initial* interview was not directly related to the model (as it naturally did not exist at that point of time), but instead covered background information on the program: its triggers, the chosen design approach as well as its uniqueness and later modifications, learnings, and exchanges with peers / externals. Concretely, the company documents for the DT program contained information on the procedure model followed for the DT, the desired target state, the as-is situation related to ongoing digital projects and competition, expected financial- and non-financial benefits, envisioned impacts on the organization, and critical success factors. Public company information was collected on the company website and the annual reports of the last three years.

For modeling, we used the online collaborative whiteboarding software *Miro*, which also offers some basic features for graphical modeling. This was a deliberate choice and constituted the best option for a proof of concept. Given the need to develop lightweight models in a collaborative and easy manner that do not conflict with the BE metamodel, EA software tools (such as *Archi*®) were not eligible in this phase. Thus, the obvious option would have been the use of presentation software such as PowerPoint. The usage of *PowerPoint* within a similar context has been researched with regards to its affordances and challenges (Ciriello et al. 2018, pp. 151–154). Miro offers the same or similar beneficial affordances. For instance, Miro provides the opportunity to order any amount of content and recombine this content in modular ways, can be used for a variety of purposes, and is digital (Ciriello et al. 2018, pp. 153–154). The increasing usage of such quite novel online whiteboarding tools due to Covid-19 and being a 'fancy' tool used by a broad set of roles within an enterprise were expected to contribute not only to creativity but also to promote acceptance of semi-formal modeling. Together with the seamless real-time collaboration feature, this ultimately led to the preference of Miro over PowerPoint. However, the usage of Miro also created some challenges, which are described in sect.7.1 below.

The model was presented (i. e., demonstrated) first to the practitioner and then used over several weeks as a running example in a course of the Executive Master program. For the practitioner, the evaluation took place simultaneously with the demonstration in the form of a semi-structured interview. The interview, conducted in German, lasted about half an hour and was recorded and transcribed. The evaluation conducted with the Executive Master students took place in the form of four focus group discussions (each with three to four students) in context of an examination feedback session after several weeks of working with the model. It is therefore fair to assume that the students had gained sufficient experience with the model. Beyond this case report and the usage in the Executive Master program, further publications and the extension to non-executive master courses are intended to communicate results.

5 Artifact description

5.1 Requirements

In the following, the general requirements (GR) (Baskerville and Pries-Heje 2010, p. 274) for a modeling method filling the white space in the DT context as well as the requirements specific to the case (e. g., especially important for it) (SR) are outlined, as well as examples of general components illustrating how the GCs could be fulfilled (where applicable for reasons of better understandability).

GR1 – Tackling complexity*:* On the general level, the modeling method needs to take into account a high complexity, both regarding the social and technical aspects of large-scale DTs in grand organizations.

GR1.1 – Tackling social complexity: The DT context affects a variety of stakeholders, having different roles and responsibilities, hierarchical statuses, business related and personal / political interests. Moreover, they are heterogeneous with regards to their educational (e. g., either technology or business, pre-experience and affinity to modeling) and professional background such as duration of staying with the organization, knowing other parts of their organization (i. e., the 'big picture') and / or organizations from former employments.

GR1.2 – Tackling technical complexity: The technical environment could also be very complex in terms of the service, process, and IT landscape / architecture. This tendentially creates technical complexities with regards to the modeling language, both in this situated case and in general. There are lots of different items (i. e., elements of a meta-model that correspond to types of real-world entities), numerous model types (each depicting one aspect of reality), and many different relationship types between the items and the models, respectively. Moreover, different stakeholders are concerned with different parts of the models and the models tend to cross disciplinary borders, while at the same time requiring deep domain knowledge in each, making their comprehension difficult especially for non-generalist practitioners. Schneider et al. (2014, pp. 2–3) identify different notions of complexity in the context of EA, that are useful to conceptualize and summarize the complexity dealt with in this context: The complexity cannot be easily (and practically useful) captured by statistical figures or metrics, instead it is *qualitative*. However, probably all involved stakeholders will conceive DTs as complex (i. e., *objective* complexity), while the degree might vary based on the pre-experience, also implying a *subjective* notion. Last, the DTs to be modeled are both *structurally* complex (i. e., trough the many entities and their relationships mentioned above) as well as *dynamically* complex (i. e., changing over time).

In this situated case, some examples for different hierarchies between the model developer (i. e., the researcher) and the model consumers (i. e., the practitioner or students, respectively) can also be found. There is an information and experience asymmetry with regards to the knowledge of the methodology and a general academic 'wisdom' in favor of the model developer. On the contrary, the knowledge of the case (for the practitioner) and general professional experience in practice (for the practitioner and the students) is more mature with the model consumers. This could represent a similar situation in an organization if the approach is intended to be used or at least initially implemented by coaching functions. With regards to the technical context, this was complex due to the properties of the DT program and organizational setting inherent in the case that was highlighted through using the BE methodology.

Besides addressing these complexities, the modeling method should lead to models that fulfill four central requirements following from the common properties of the use cases as described above.

GR2 – Highlighting change: First, the method should bring in a change perspective, recognizing that the use cases do not take place on a green field, but instead in an existing organizational environment. Thus, the as-is and to-be states need to be clearly depicted and changes highlighted on the appropriate level of detail depending on the needs of the model developers and consumers.

GR3 – Emphasizing relationships: Second, the model should create and highlight relationships between the models (i. e., create coherency) opposed to approaches focusing only on specific aspects. While this has already been noted by Winter (2011, p. 6), this has become increasingly relevant through the 'inflation' of strongly local models (e. g., as canvases) becoming popular (Avdiji et al. 2020, p. 696; Sandkuhl et al. 2018, p. 70).

GR4 – Enable collaboration: Third, it needs to be collaborative to reflect the work mode and the (use case specific) purpose (i. e., no documentation just for compliance reasons). This should also facilitate acceptance through being intellectually compelling ('making fun').

GR5 – Establishing enterprise-level view: Last, the enterprise-level view needs to be accomplished by promoting consistency and re-use of organizational entities. Consistency in general can take the form of internal consistency (i. e., intra-model and inter-model) and external consistency (i. e., enterprise-related and market-related) (Kautz 2022, p. 33), whereas the latter form is especially relevant for the enterprise-level view. Consistency implies rather 'superficial' aspects such as using a shared terminology for the description of organizational entities (e. g., product names) and also more sophisticated aspects such as not making design decisions that are not compliant with the normative restrictions of an enterprise (e. g., with its overall business objectives) (Kautz 2022, p. 33). Thus, the method should provide a meta-model specifying (potentially hierarchically decomposable) entity types of enterprise-level relevance and also incorporate consistency-producing and -checking activities into the procedure model, in order to ensure that all relevant entities of the respective type are considered.

Another set of requirements follows from the different purposes of the model in this situated case that can be traced back to the use cases.

SR1 – Reframing the DT endeavor: To serve as a tool for collaborative analysis and reflection, the model needs to offer a correct and 'quite' comprehensive depiction of the case details, to be at all able to identify gaps and optimization possibilities. Moreover, it needs to enable further insights beyond the facts (that are already known by the practitioner) and thus serve as an 'eye-opener'. Therefore, it needs to offer a meta-model that focuses on the essential elements and their relationships and a notation that presents this information in an adequate way leading to a novel insight by the model consumer. Verbal reframing of problems has in the case of designers (e. g., engineers) been reported to influence the number of ideas generated (Silk et al. 2021), reframing of the (management) problem through the conceptualization of the meta-model might bring similar benefits. Another cognitive / psychological motivation is given by the *Cognitive Fit Theory*, roughly postulating that how a problem representation highlights the information types necessary for problem solving (e. g., visually) impacts performance (Vessey and Galletta 1991, pp. 65–67). Thus, visualizing the content of the business-to-IT-solution is expected to be beneficial, for example, if the structure of an endeavor should be assessed (problem) and a notation that highlights the relationships (information types) between the different projects is used.

SR2 – Using a deliberate notation: For research, the method needs to incorporate a deliberately chosen notation that reflects the other purposes the model is used for.

SR3 – Exploiting software affordances: Moreover, the method should consciously exploit different software affordances / features.

SR4 – Being easily understandable: For teaching, easy understandability of the models in a limited timeframe was important and trade-offs between the richness of the case and focusing on the most important elements need to be addressed. For the modeling method this implies, that it should provide prioritization and aggregation mechanisms. Moreover, the modeling language should not be too complex to facilitate acceptance, e. g., by using only a limited number of symbols and relationship types.

SR5 – Switching between general and situated case: On top of that, showing both general relationships as well as their concrete application was necessary, implying for the modeling method or rather the software tool to enable making meta-level comments to the developed models.

5.2 Resulting modeling method

When generalizing the *activities* presented in sect.4.2, the method contains the phases of clarifying the modeling context, selecting model types, collecting data, analyzing data and iteratively developing the model, possibly evaluating the model, and using the model. While these phases are identical for all use cases, the concrete content of these phases will be different, especially for the later phases.

Clarifying the modeling context comprises defining the use case in which the model should be applied, identifying the stakeholders involved in it, and gathering their superordinate goals and requirements on the model, esp. their information needs (cf. de Kinderen 2017, p. 225), following from the use case. Then, the model types satisfying the requirements and providing the necessary information are selected. Data are gathered via a variety of sources (e. g., interviews, secondary documents) and especially already existing models (both within and outside the organization, i. e., reference models) are considered. However, in the case of the reflection tool, the data sources (e. g., presentations about the business-to-IT-solution implemented by the endeavor) are already existing and in turn determine the necessary model types. The data are then analyzed with regards to the meta-model entities' instances they contain that are the basis for the model, which are then carefully supplemented by the relationships between them and the models. The sequence of developing the partial models is mainly a top-down one from the strategy to the IT layer oriented at the BE framework (Winter 2011, pp. 70–84). Besides the iterative nature of model development, i. e., with different feedback cycles within the model developer team and with the model stakeholders, one might also include a more concluding evaluation, such as before the model is presented to important committees or a broader audience. The usage phase of the model in the reflection use case then contains the presentation of the model to the (whole) DT management team, its discussion, and the documentation of insights.

The main *results* are, besides the model, a documentation of the stakeholder requirements (e. g., a stakeholder perspective catalogue as proposed by de Kinderen 2017, p. 226), possibly a folder structure for comprehensive data sources, and the documented results from the usage phase.

The main *technique* is the modeling technique (besides this sect.cf. the next sect.for the modeling language) and supplementary techniques for data collection, analysis, and requirements engineering can be used.

The *information / meta-model* underlying the modeling technique is based on an extension of the BE meta-model (Österle et al. 2007, p. 193; Winter 2011, p. 19), complemented by the virtual decoupling meta-model (Aier and Winter 2009, p. 156) as well as Osterwalder's (2004, pp. 42–102) *Business Model Ontology*, complemented by the *Value Proposition* concept (Osterwalder et al. 2014, pp. 6, 8–9). The meta-model itself is illustrated in fig.2, the entity types that are instantiated in each partial model are listed in Tab. 3. As we integrated partial models on the meta-model layer through meta-model slicing (Bork and Alter 2020, p. 9), the meta-model was an important lever to emphasize relationships (cf. GR3). However, the depiction of the meta-model does not completely explain all the relationships between the models for several reasons: First, space constraints and the graphical complexity that can be understood and maintained by a user, set limits to the number of entity types and their relationships which can be depicted in the meta-model. Therefore, only the most important relationships between entity types are shown and not all specializations of entity types can be displayed (e. g., for resources). Second, for entity types that are undergoing changes / different stadiums (e. g., a *process output* that is modified / extended to at some point become a customer service), the meta-model only provides two separate, but linked entity types. Third, the meta-model cannot depict how certain configuration / combinations of design decisions, i. e., selected instances of entity types, at an earlier design stage affect / limit the set of consistent design options later. This is especially the case for 'attribute-like' entity types. For example, the Model of Competitive Positioning for MedComp already 'prescribed' on the strategic layer that the new service should be based on data which is integrated not only from devices of its own brand, but also from other vendors. Thus, one expects that at the organization layer activities are described to accomplish this. Similarly, the meta-model only poorly reflects 'emergent properties' of certain designs, e. g., how a process contributes to the non-functional requirements posed by the customer. Forth, some entity types (e. g., cost drivers) provide a new perspective on the business-to-IT-solution and therefore can be related to more entity types (as well as their combination), which cannot be depicted in the meta-model. However, these relationships are nevertheless highly relevant for further planning as well as communicating the business-to-IT-solution. The just mentioned limitations of the meta-model create the need to supplement the syntactical rules it establishes through semantical rules specified in natural language (Frank and van Laak 2003, p. 21; Vogel et al. 2009, p. 267). Taking the examples of the Business Model Canvas and Value Proposition Canvas, the books by Osterwalder and Pigneur (2010) and Osterwalder et al. (2014) provide an extensive set of such semantical rules through giving instructions for correct use of these canvases.

Finally, regarding the *roles*, the authors refer to Stirna and Persson (2012, pp. 663–664) for a more differentiated view on model developers and consumers.

5.3 Resulting model

Following the definition of a semi-formal model introduced in sect.1, the resulting model can be called a semi-formal one. The contingent of symbols (i. e., notation) is comprised of a subset of all available so-called *sticky notes*, other forms, and arrow types in Miro, that graphically represent instances of the method's meta-model entity types. A depiction of the notation can be found in fig.3. The notation follows from the requirements presented in sect.5.1. Icons used in the upper right corner of each partial model characterize it based on the level of the enterprise which it concerns (cf. GR5) and whether it depicts the as-is or to-be state of the enterprise (cf. GR2). The color of the sticky notes denotes whether the instance of an entity type is already there (light yellow) or new (blue, i. e., as a consequence

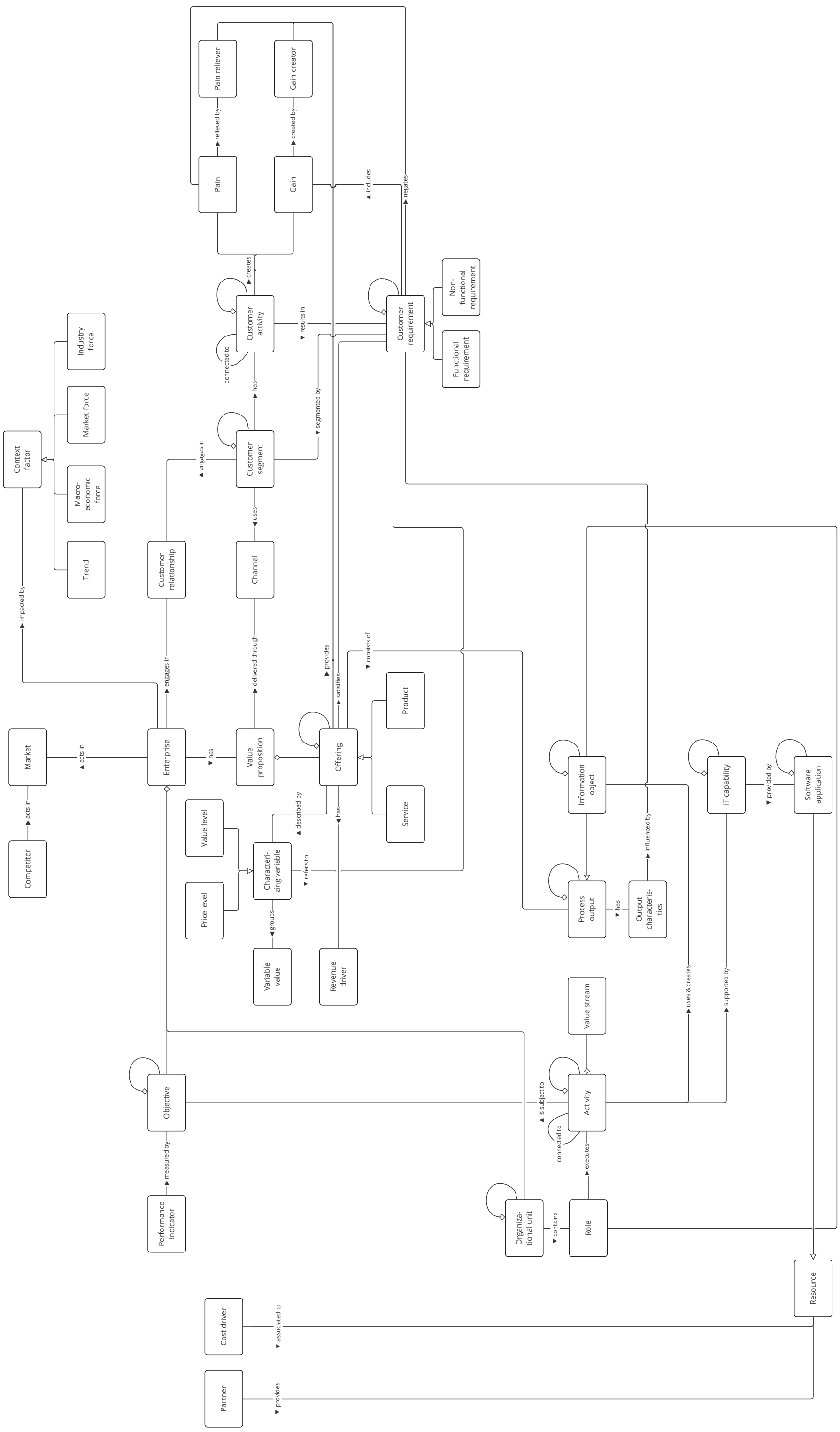

Figure 2: Meta-model. ***Note:*** *The meta-model is modeled oriented at UML classes using association, generalization, and aggregation relationships (Object Management Group 2017, pp. 718–723). The little triangles indicate the reading direction of the relationship labels. Please keep in mind the limitations of the meta-model discussed in sect.5.2. The meta-model is based on Aier and Winter (2009, p. 156); Österle et al. (2007, p. 193); Osterwalder (2004, pp. 42–102); Osterwalder et al. (2014, pp. 6, 8–9); Winter (2011, p. 19).*

	Business Model Canvas	Business Model Environment Analysis	Business Ecosystem Model	Model of Competitive Positioning	Value Proposition Canvas	Customer Journey Model	Model of the Product / Service Portfolio	Objective Map	Value Stream Architecture	Frontstage Process Model	Backstage Process Model	Process Performance Analysis	Task / Role Analysis and Organization Structuring	Information Flow Model	Information Object Model	Capability Map	IT System Landscape
Enterprise	*		x				*	*	*				*		*	*	*
Market		*															
Context factor		x															
Competitor			x	*													
Customer segment	x		x		*	*			x	*							
Offering	x		x	*	x	x	x		x								
Channel	x																
Customer relationship	x																
Activity	x								x	x	x	*	x	x			
Resource	x		x							x							
Partner	x		x							x							
Revenue driver	x																
Cost driver	x																
Customer activity					x	x				x							
Pain					x	x											
Gain					x	x											
Pain reliever					x												
Gain creator					x												
Characterizing variable				x													
Variable value				x													
Customer requirement												x					
Objective								x				x					
Performance indicator								x									
Process output												x					
Output characteristics												x					
Organizational unit													x				
Role													x				
Information object														x	x		
IT capability																x	
Software application																	x

Table 3: Entity types as they are used in the partial models. ***Note:*** *Instances of entity types directly shown in the model are denoted by a ‘x’. Instances of entity types, which the respective model in total relates to, are marked with a ‘*’.*

Figure 3: Notation used in the developed model

of the DT). Different types of arrows represent the different relationships between entity type instances and the partial models, respectively (cf. GR3).

With one exception, the developed *total model* (in the following, just referred to as ‘model’ for reasons of simplicity) mainly consisted of *partial models* (i. e., aspect models) based on existing *model types* (i. e., templates; source to be found in fig.4) that were ‘filled out’ by the researcher and practitioner during method application. Only the capability map was a refinement of an existing reference model, using domain-specific maps by LeanIX (2022).

As depicted in fig.4, the model sorts the partial models along three dimensions, whereas only two can be shown based on spatial allocation. The first dimension refers to the architectural layers (cf. van Gils and Proper 2018, p. 262) and classifies the partial models based on whether they show aspects of the business-model within the business-to-IT-solution (*strategy layer*), the structural and procedural organization (*organization layer*), the IT capabilities (*alignment layer*), or IT implementation (*IT layer*) (Winter 2011, p. 27). The second dimension captures whether the model describes the current situation (i. e., as-is) or desired target state (i. e., to-be) of the specific aspect of the business-to-IT-solution. The third dimension characterizes whether the partial model depicts an aspect in a way that is relevant for the business-to-IT-solution and / or the whole enterprise. This can be a consequence of the circumstance that an analysis can logically only be done on a certain level, that specific parts of a partial model are only relevant for the respective level, or that a more detailed / concrete or aggregated / abstract view is needed.

fig.5 shows an exemplary partial model based on the *Value Proposition Canvas* (Osterwalder et al. 2014, pp. 8–9) that was developed quite early to validate that the services that should be newly offered by MedComp fulfill customer needs. It is read from the right to left and, first, lists the activities a typical customer needs to perform. Given this, it identifies the negative and positive feelings (pains and gains, respectively) that the customer has while carrying out the activities. Then, at the right of the rectangle, concrete features of the new services are identified that address these

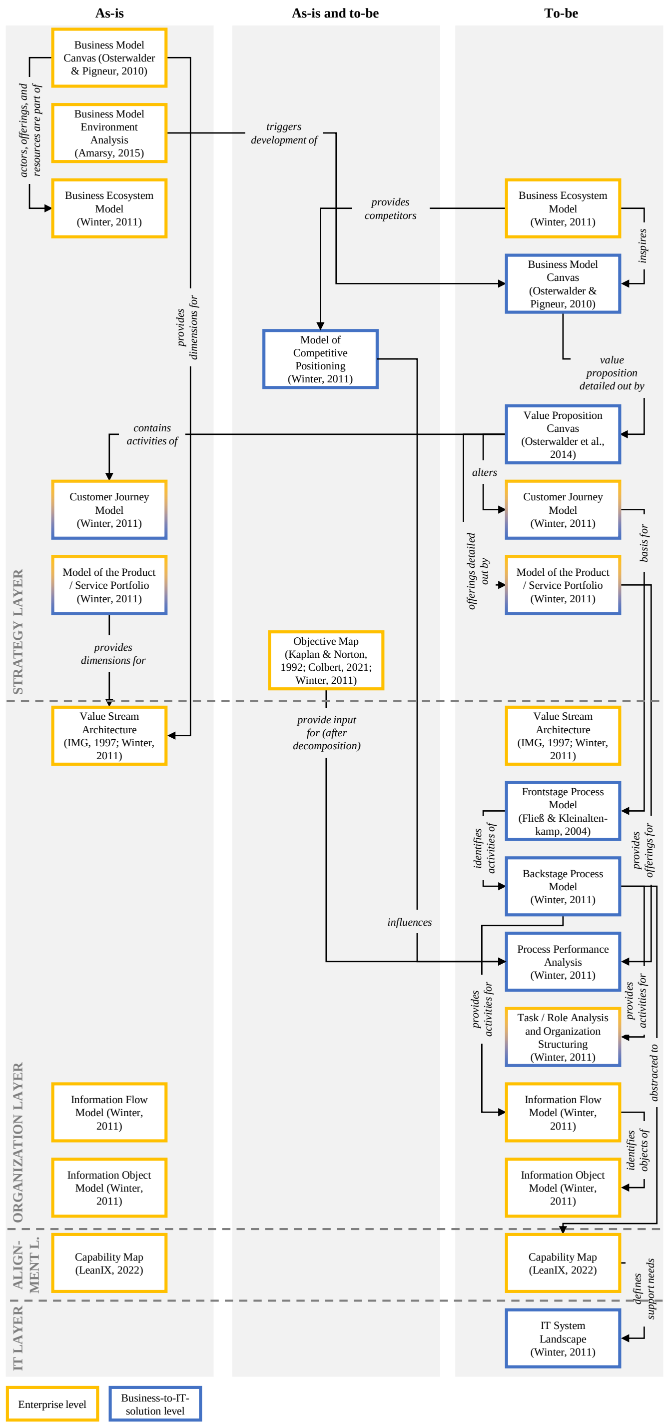

Figure 4: Schematic representation of the partial models and their most important relationships. ***Note:*** *For reasons of readability, relationships between partial models existing in the as-is and to-be states are only shown in one of the two. The templates for the partial models are based on Amarsy, N. (2015); Colbert, B. (2021); Fließ and Kleinaltenkamp (2004, p. 397); IMG (1997); Kaplan and Norton (1992, p. 72); LeanIX (2022); Osterwalder and Pigneur (2010, p. 44); Osterwalder et al. (2014, pp. 8–9); Winter (2011, pp. 99, 101, 102, 106, 111, 115, 117, 119–120, 180, 183); and the layering is based on Winter (2011, p. 27).*

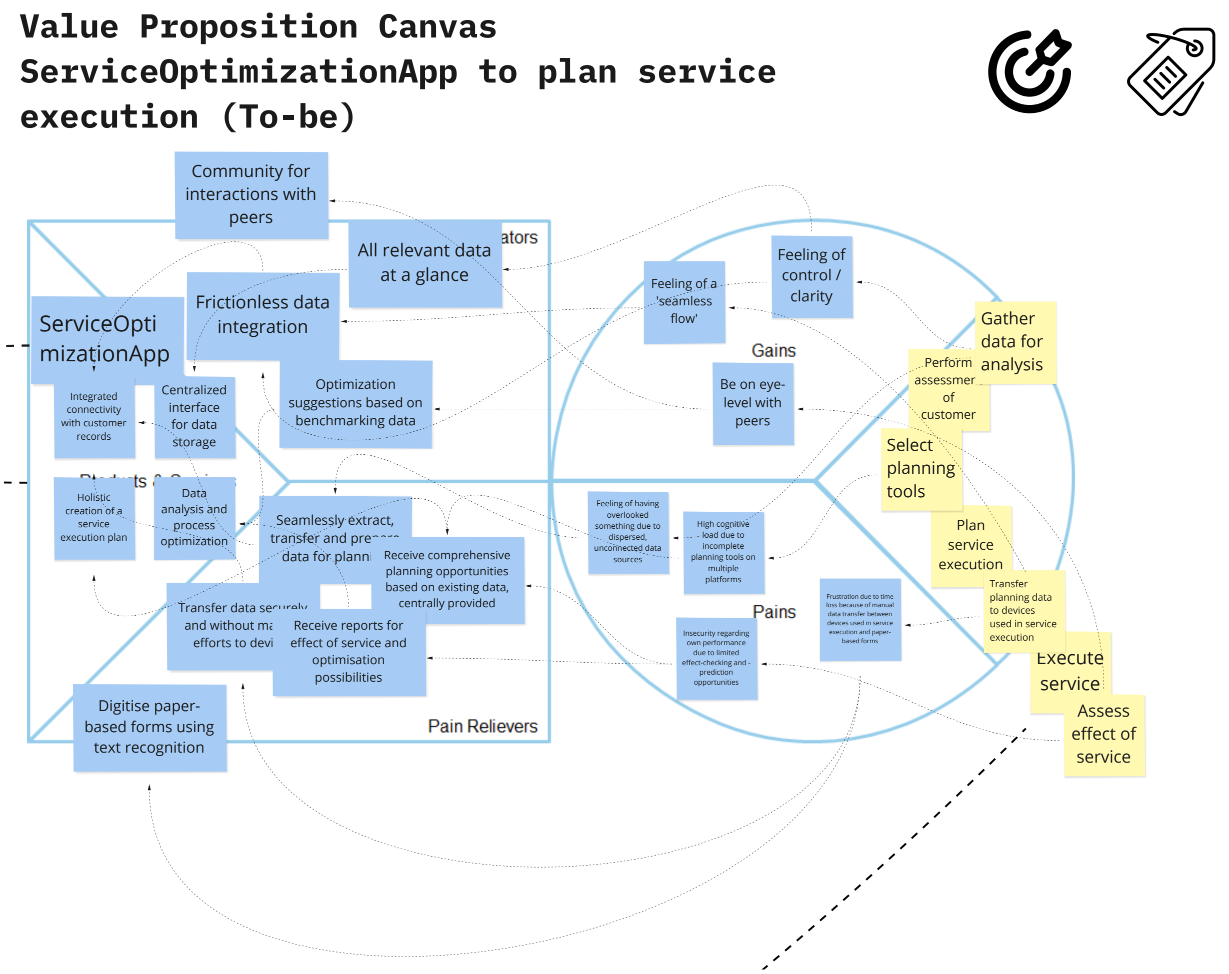

Figure 5: Value Proposition Canvas (to-be) as one exemplary model (modified for reasons of confidentiality). ***Note:*** *The Value Proposition Canvas template originates from Osterwalder et al. (2014, pp. 8–9).*

pains and gains (pain relievers and gain creators, respectively). It is important to note that – opposed to a Value Proposition Canvas developed in a quick, creative ideation session without many restrictions – here it is required that each pain reliever / gain creator is associated to at least one pain / gain, respectively. Otherwise, this implies that the service contains a feature not adding value for the customer, and accordingly should not be implemented. This is an example for a (syntactical) rule imposed on the originally informal canvas. The fulfillment is checked through the notation requiring drawing arrows between the sticky notes. However, referring back to the discussion of syntactical and semantical rules in sect.5.2, it should be noted that this rule is not sufficient to guarantee that the pain relievers / gain creators *indeed* provide utility to the customer. Supplementary instructions are needed to ensure this. Last, the very left part of the canvas bundles the features into the concrete services that are offered by the company and this way ensuring the attractiveness of the services for the customer. The Value Proposition Canvas was the basis for the development of the to-be *Customer Journey*, which is improved through the new services (e. g., some previously manual steps of the customer journey became obsolete).

fig.6 shows a *Front Stage Process Model* based on a *Service Blueprint* (Fließ and Kleinaltenkamp 2004, p. 397). This model was used to check whether the activities (i. e., processes and their steps) that were installed for the new business-to-IT-solution are aligned with the customer's activities (i. e., the customer journey), e. g., that the right process outputs are available at the right point of time. On the first layer, the customer journey is copied from the corresponding previous partial model. On the subsequent two layers the activities that need to be performed by the app, either visible for the customer or not, respectively, are identified. Then, activities that must have occurred in preparation (i. e., no execution while using the app, but before) are identified. Finally, the most important resources are identified that need to be available to enable carrying out these activities. Like in the Value Proposition Canvas, sticky notes without any connections to other sticky notes are not allowed. The Service Blueprint makes it possible to differentiate activities based on the customer's needs, i. e., what the customer wants to have and at which time. This perspective is not needed in the backstage process model, which was for example used to derive the necessary roles for delivering the service. Thus, to reduce complexity, the activities were put on a single layer in the backstage process model of the app.

6 Demonstration and evaluation

6.1 Model usage

Concretely, the practitioner used the model in two ways. Given the retrospective nature of the model development (i. e., for reflection and improvement after an initial phase of DT program design), not all parts of the model equally triggered changes on the business-to-IT-solution. However, as work was ongoing on the IT-related entities (while, e. g., the design of the value proposition was already finalized), the capability map and IT system landscape directly influenced ongoing work. Second, the model provides a valuable basis for future activities.

The students used (and later cohorts are still using) the model to, first, see at a real-life and detailed example which design decisions must be taken to design the future state of a 'digitally transformed' company. Second, they also used it as an inspiration for how to structure the presentation of and the notation for their own case studies' models. The cases the students work on are mostly taken from their job-related digital business innovation endeavors.

6.2 Results of the internal evaluation

As described in sect.4.2, the internal evaluation was based on the (sub-)criteria for model quality proposed by Becker et al. (2012, pp. 32–36), to which we refer in the following.

The first one is semantical as well as syntactical *correctness*. Given that the modeling language

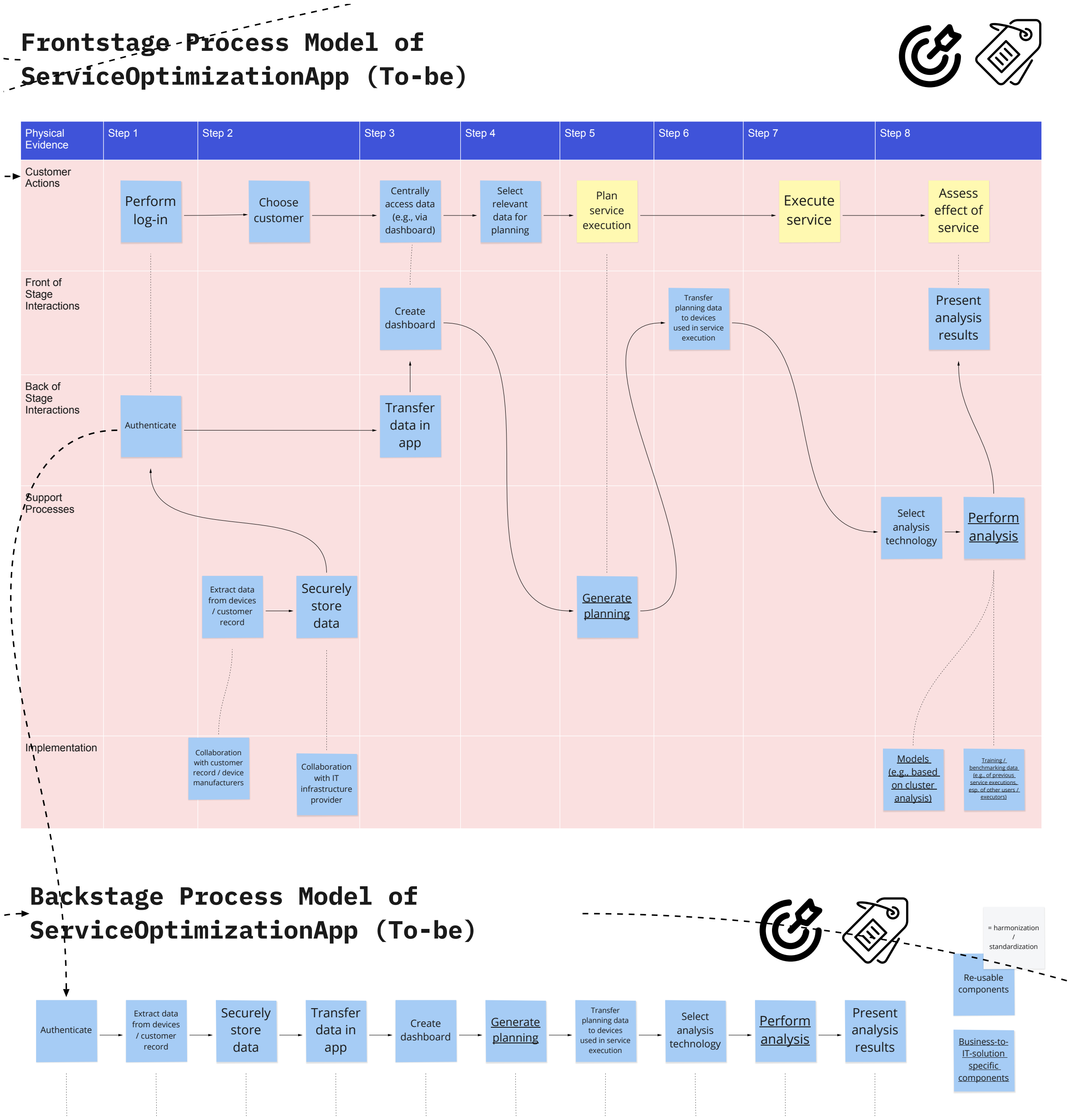

*Figure 6: Frontstage Process Model (oriented at the to-be customer journey) and Backstage Process Model as exemplary partial models (modified for reasons of confidentiality). **Note:** The Service Blueprint template as the basis for the Frontstage Process Model is content-wise based on Fließ and Kleinaltenkamp (2004, p. 397) and provided by Miro (2022).*

was specifically developed for this model, the criterion of syntactical correctness does not seem applicable. The semantical correctness is determined by the practitioner feedback in the next section. *Relevance* recognizes that not all real-world entities and relationships are relevant for the model consumer and can be abstracted. This can be achieved through explicating the modeling aims at the beginning, which was also done in this case (cf. sect.2) and influenced the selection of the model types and the modeling language. Relevance can first be decomposed to internal minimality, i. e., modeling all aspects that are relevant in the real world, which was achieved through asking the practitioner and students for missing elements. Second, it also contains external minimality, i. e., modeling only aspects that are relevant in the real world, which was achieved through identifying such aspects from the perspective of the practitioner or students, respectively. Besides these effectiveness-related criteria, another one is *economic efficiency* building on the minimum and maximum principles in economics. Efficiency was achieved through considering the additional (i. e., marginal) didactic value when refining the model. Moreover, the reference capability maps served to convey the procedure to derive capabilities from process models to the students (given aim) with less efforts than developing a capability map from scratch (minimized effort). *Clarity*, consisting of easy readability, aesthetics, and comprehensibility (cf. evaluation in the next sections) can be enhanced through an appropriate hierarchization, layout design, filtering, and consistent terminology (within models and between the models and the real world). The layout design was beneficial through an infinite presentation space (due to Miro) and improved by highlighting similarities through modeling structurally similar components (e. g., parts of a partial model) in the same way. Hierarchization was implemented content-wise via partial models that provided a 'deep dive' into another partial model (where appropriate) but appeared to be in general less relevant for a DT endeavor in an early phase that rather does not require analysis on different levels of detail. Filters were not directly implementable in Miro. The consistent terminology could be implemented through, e. g., adopting the wording of the case company where possible. The criterion of *comparability* was not applicable in this case. *Systematic structuring*, i. e., ensuring the input-output-relationships between the partial models to achieve inter-model consistency, is already a core feature of the modeling method (cf. sect.5.1) and was thoroughly implemented, for instance with the information flow describing the usage of the business objects and the information model describing the components of exactly the same business objects.

6.3 Results of the practitioner's evaluation

Due to the assumed novelty of the modeling method, the focus of the semi-structured interview questions was – besides correctness and clarity of the actual model (cf. previous section) – on the modeling method's relevance (esp. importance) and utility, both in general and in comparison to other approaches (cf. Sonnenberg and vom Brocke 2012, pp. 393–394; Venable et al. 2012, p. 426).

The *correctness* of the model was assessed to be "very good"[3] because the (quite comprehensive) content from the slides was described as "really well brought together" and "matched to the right parts of the [total] model". With regards to *clarity* in terms of readability and aesthetics, it was mentioned that having the model in Miro is a "really good thing" because it allows to navigate through the model. In terms of readability, the many partial models caused a missing "red thread" or "cookbook". Instead, the partial models should be clustered into a few overarching, big steps or phases, maybe implemented through filters. A thematical clustering was suggested as an extension. Remarking that the model development in this case was more a "retrospective", the practitioner confirmed that the modeling method and model are *useful* and he will use (parts of) them in future. Moreover, he noted that he could

[3] Phrases in quotation marks refer to literal statements of the practitioner (translated to English).

have carried out the content-wise similar activities from the actual, past setting-up of the DT program more easily given he had known the modeling method and model before. Moreover, he said that for "all greater undertakings you carry out in a company, you need a method or a model". These provide clarity about and unify the concrete steps, tools (i. e., the partial models), and terminology to tackle an issue, against the background that bigger teams are naturally not aligned by default. The perceived utility was exemplified by the practitioner having passed on the model internally, where now deeper work was triggered on the capabilities and IT architecture, which should also be considered more in future endeavors. Regarding the *utility in comparison* to other methods used for analysis and design in the DT management context, which were in this case a strategic staircase, an operating model (defining processes, governance, organization, tools, and strategy), and an implementation plan, the modeling method was assessed to be "very complete" and "very broad". Consequently, it is also more complex than the other approaches, calling for guiding users on how the different partial models are related, which aspects they cover, and when and how to use them.

However, if the practitioner would apply the method, the practitioner would not decompose the model, but instead first define when to use the partial models and allocate them to colleagues from different domains, depending on whose knowledge is needed and for whom the respective model is relevant. Related to this, the modeling method and model would rather not be used in context of lessons learned, given that not all partial models are needed for more specific questions. In general, the approach was also seen as a tool for strategy and solution development / design rather than for project and program management – because the latter tasks are more concerned with ensuring effective, efficient, and timely solution delivery. However, for the development phase, modeling was considered to be "very important, because if you don't model it neatly at the beginning and think it through, you will later have a problem in the implementation", also leveraging the creative inputs of multiple functions to achieve a "clever" product or service at the end.

6.4 Results of the students' evaluation

Guiding questions for the focus group asked for the perception of the situated model's clarity (cf. sect.6.2), the modeling method's utility in the professional context of the (part-time) students (which all are employed and have many years of managerial experience), and especially valuable and missing features of the method.

The model itself in general was assessed to be useful to convey the course contents, which became apparent through the wish to extend it to all models shown in the course – while readability and aesthetics should be improved. Along a similar line, Miro was considered to be very suitable, but the transfer of the Miro results to well-designed presentations was considered as a problem. The modeling method was found to be a well-suitable "model and method box"[4] for the collaborative analysis and first end-to-end- design decisions in common business innovation projects, also in a transformational context. Regarding the method's results, the models dealing with the alignment and IT layer (capability map and application landscape) were identified as very valuable because they provide compatibility with the existing business and IT architecture as well as a basis for implementation discussions. However, support for organizational design was considered insufficient, as it was called to be a "crucial point" in many DT endeavors and at the same time less easy to grasp than the strategy aspects receiving most attention (cf. fig.4). Moreover, the method's results should also be complemented by providing modeling variants, i. e., alternative models for the same aspect. A missing method activity was a support for selection of the partial models, i. e., highlighting under which circumstances a specific model has which advantages and disadvantages. Besides that, the modeling method was assessed to be quite complete since no participant added a further method component or model template that was considered to be missing.

[4] See previous footnote, which applies analogously here.

6.5 Return on modeling effort

The expected RoME was already described in sect.2, the achieved return as indicated by the evaluation just presented. While the highest efforts were invested by the researcher, less by the practitioner (e. g., time for the interviews) and the least by the students (e. g., actively participating in the focus group), the latter ones were primarily benefitting from the creation of the model and the modeling method. However, it should be noted that the researcher also benefitted in his role as a scientist since the case offered the chance for scientific inquiry and in his role as lecturer since the model facilitated teaching.

Note that the understanding of what constitutes a positive RoME in this case study is quite special. An emphasis was not placed on, e. g., the fulfilment of syntactical rules, in order to support easy and error-free automation. Instead, a focus was placed on the four central requirements explained in sect.2. Only when fulfilling these requirements, the needs of the stakeholders, which are central to the concept of RoME (de Kinderen 2017, p. 226), can be satisfied. The practitioner, the students, and the researcher were interested in increasing the effectiveness of DT endeavors and accordingly selected modeling paradigms and the concrete model templates as well as developed the modeling language. Just loosening restrictions from existing semi-formal modeling languages and apply them on a non-detailed level would not have been sufficient for that purpose. Moreover, realizing this RoME was not achieved by having no rules at all during model development. Instead, coherency and (semantic) consistency had to be considered, which are central to ensure that the real-world-entities that are designed based on these models deliver the desired benefits, increasing the effectiveness of the DT endeavor.

7 Discussion and conclusion

7.1 Reflection

Several lessons can be learned both from the development of the general modeling method and the presented situated model. The lessons learned regarding the general modeling method have already mainly been integrated into the requirements outlined in sect.5.1. As the lessons learned in context of the development and evaluation of the situated model were neither known upfront nor the consequence of a dedicated requirements gathering but emergent, they can be used for the future further development of the method.

One lesson learned originating from the development of the *general* modeling method concerned the overwhelming broadness of the real-world that should be modeled as perceived by model consumers, i. e., the holisticness, the variety of models needed, and ensuring the relationship between them. A second one is that this was intensified through missing comprehensive (i. e., total) models and missing software tools that also enable a holistic presentation. A third one relates to the point that the models cross discipline borders and often also go beyond the competences of individual model consumers. These lessons were also already outlined in Kautz (2022, p. 80).

Several, more detail-level lessons learned were also gained based on the internal collaboration and evaluation by the researchers in this *situated* case.

There were different aspects that were specifically challenging and related mainly to our research method and setting, i. e., not to the problem of supporting DT through modeling as such. One was identifying and explicitly showing the consistency of the overall business-to-IT-solution in parallel (and partially ex-post) to the sub-projects that actually designed, engineered and implemented the DT. For future research projects we therefore aim to already participate in the design phase of the DT.

Another challenge was comprehending the case given its industry-specific terminology and processes that are difficult to understand for a layperson. On the other side there were some ontological misunderstandings between the practitioner and the researcher, e. g., on the definition of a capability, which we thought of as an IT capability and the practitioner as an employee capability / skill.

Using the meta-model for a systematic alignment of the practitioner's and our understanding of key terms at the beginning of the project might have been helpful.

Last, adopting a user-centric (i. e., concerning the practitioners and students) perspective constituted a challenge. In the case of the students, this was amplified through their heterogeneous organizational, educational, and professional backgrounds. Thus, the situations (i. e., use cases) in which they personally use models will be heterogenous. By contrast, at the same time, there was the need to teach a generally applicable approach instead of providing support specific to their situated problems.

Moreover, it became apparent that modeling change in context of this DT was not trivial but required a differentiated approach to separate between and visualize the as-is and to-be business-to-IT concepts (cf. van Gils and Proper 2018, p. 263). On the one hand, as-is and to-be is not black and white, as there might be entities that have already existed before and are only changed and the allocation of a partial model to as-is and to-be is not always possible. On the other hand, the notation needs to offer differentiated ways of showing the as-is and to-be entities, given that a simple graphical modeling tool like Miro is used. In concrete, this is color coding (to be used within partial models) and the spatial allocation of as-is and to-be models in different areas of the modeling space (e. g., swim lanes).

Similarly, and also related to a distinctive property, relationships between the partial models were expressed in a differentiated way, using arrows, re-use of the same sticky notes, and intermediate models (e. g., comparisons) showing the process to arrive at the other partial models.

The differentiation between the analysis and design levels (i. e., business-to-IT-solution and enterprise) also required both a within-model highlighting through the notation and a classification implemented via icons for the whole partial model.

Furthermore, besides the decision whether a certain partial model can be logically developed for the respective level, the choice whether to actually do this also needed a prioritization with regards to the didactic value it would provide.

An additional observation was that there is the need for orientation at popular visualizations and terminology as well as the challenge to provide a meta-model that is up to date.

As already noted in sect.4.2, there were also lessons learned with regards to using Miro as a software tool. Establishing consistency was challenging as Miro does neither support the re-use of elements created in a previous partial model nor by default the re-use of already existing modeled entities (e. g., in an EA software tool). Moreover, relationships between the models are not highlighted automatically. Filters that lead to stakeholder-specific models were also not implemented. These are some examples for affordances that Miro is missing, additional ones, e. g., not implementing and enforcing a syntax, could become relevant when the modeling method should be applied by laypersons as opposed to researchers.

Several lessons learned were also derived based on the practitioner and student feedback. The students' 'model and method box statement' (cf. sect.6.4) as well as the practitioners perception of completeness seem to imply that the distinctive characteristics of the modeling method mentioned in sect.2 are recognizable.

That the support for the organizational design was perceived as incomplete by the students was quite astonishing, given that models regarding the architecture, frontstage, backstage, performance, information supply of processes and models of organizational structures were provided (cf. fig.3). Possible starting points here could be including more detailed guidelines (i. e., activities) on how to develop processes or deriving the organizational structure with an emphasis on concrete competencies and skills, maybe drawing on insights from the process engineering / business process management and human resources literature, respectively. The challenge here is finding the aspects that are relevant for the decision whether and how to proceed further with the development of a business-to-IT solution, without going too much into detail in this early phase of DT endeavors.

Also noteworthy is that both the practitioner and the students emphasized the importance of the capability map and IT landscape, highlighting the need for an enterprise-level view.

The demand for a model selection procedure expressed by both the practitioner and the students and alternative models raised by the latter fits well to our assumption of a heterogeneity of model consumers, which was also indicated by the practitioner, and to the related research stream of modeling in the EA context, differentiating between the different *viewpoints* that stakeholders have (Kurpjuweit and Winter 2007, p. 146). This relates to the expressed preference of the students for a software tool allowing to create presentations appropriate to the target group (i. e., their viewpoints) as well as a more readable and aesthetic presentation of the models in general, implying besides a gradually, potentially animated presentation of the models (which would also accommodate the stepwise presentation desired by the practitioner), filtering for different model elements. A precondition for such filters is a meta-model, which cannot be implemented in a simple graphical editor / whiteboarding software like Miro. The challenge might be developing a tool as easily and commonly adoptable as such a whiteboarding tool while implement a (rudimentary) meta-model allowing the creation of viewpoints (e. g., based on an element's layer, relative importance, etc.) and facilitating presentation.

With regards to the use case as a reflection tool, the main reflecting insight by the practitioner seemed to be related to the completeness of their design work in comparison to the partial models, detecting that the capabilities and IT architecture were insufficiently considered. However, the possibilities to reflect on the set-up of and collaboration within the endeavor enabled by a comparison with the conceptual relationships was not mentioned. Moreover, the modeling method was assessed to be rather relevant in the design phase of the DT, which was treated as a clearly distinct phase of the implementation. Given the more agile, iterative approaches at least used in projects of today, this raises the question whether design (re-)work should not also be part of the implementation phases as well, what, however, goes beyond the scope of this research. Summarizing, the use of semi-formal modeling for endeavor management might be either irrelevant or its possibilities not enough convincingly / explicitly communicated in context of this research.

7.2 Contribution and future research directions

This research contributed to practice by showing how semi-formal modeling could be used as a value creating tool in DT endeavors and presented an exemplary, concrete model to be used as an orientation. To research, it contributed through theory-inspired descriptions of experiences / lessons learned and a rigid evaluation that highlight further research opportunities that are detailed out below. For both audiences (i. e., practitioners and researchers), the attempt to integrate canvas-based and 'classical' semi-formal models, exemplified through the meta-model in fig.2 and the overview about the partial models in fig.4 is expected to be a useful approach providing overview and guidance in situations where, first, trends and competitors force a company to transform (i. e., the company is in a rather adopting / following position). Second, this DT needs to include the extension of the product portfolio through 'smart', digital, services. Third, the demand for and the novelty of the skills must justify making (and thus modeling) changes of the organizational structure.

While the intertwining purposes for which the model was used facilitated comparisons, it limited the insights through going more into breadth than depth. For instance, the same use case could have been investigated in multiple organizations or several use cases could have been investigated instead, improving the generalizability of the findings.

This practitioner case highlights several avenues for further research. First, one might want to refine the method as such, also considering the lessons learned from the previous sect.as well as the literature presented in sect.3. A first step could be done by further analyzing the practices underlying the use cases. On this basis, the method's activities could be detailed out with different ‘paths’ (cf. situational method engineering), also considering the specialized activities implied in the work of Babar and Yu (2019, p. 7). Especially, the model selection procedure could be formalized. Previous research has already investigated the situational, value-based bundling of method fragments and modeling languages, respectively (de Kinderen 2017, p. 222; de Kinderen and Proper 2013, p. 1469), which was not the primary concern here. However, this idea could be extended to the value-based selection of model types in this method (i. e., based on business (information) requirements, that in turn are rooted in the use cases that should be investigated in more detail). Also not intended here was the combination, but instead the development of a lean, possibly slightly adaptable, and unified modeling language across all partial models and use cases to accelerate adoption and allow for easier model transformation between use cases. One could develop the modeling language in a more rigid way, e. g., leveraging approaches such as the one by de Kinderen and Ma (2015), more deeply examining the recommendations by van Gils and Proper (2018, pp. 267–270) and McGinnis (2007, p. 141), and also orient the notation at insights from cognition, creativity, and learning theory. Second, going more in the software tool direction, the affordances of tools to support modeling in the use cases could be further investigated, incl. the research on already existing tools and the ones to be developed.

This case report presented a semi-formal model as an instantiation of a novel modeling method, that is collaborative, holistic and integrative, enterprise-level oriented, and change-capturing to support decision-making and change management practices in the context of DT endeavors. We also expect it to be projectable to smaller-scaled digital business innovation endeavors given that these and DT endeavors have changes with the same depth (see sect.2), while it might be even ‘easier’ applicable to the first due to their reduced complexity as a consequence of their narrower breadth. Since the method integrates popular canvases, it shed the light on treating them as semi-formal models, which also opens up further research opportunities. The relevance of the developed and similar semi-formal models was shown and reasoned, calling for an intensified use of models at work, causing practitioners like ‘ours’ to say: “It was fun”.

References

Aier S., Gleichauf B. (2010) Application of Enterprise Models for Engineering Enterprise Transformation. In: Enterprise Modelling And Information Systems Architectures Journal 5(1), pp. 58–75

Aier S., Winter R. (2009) Virtual Decoupling for IT/Business Alignment – Conceptual Foundations, Architecture Design and Implementation Example. In: Business & Information Systems Engineering 1(2), pp. 150–163

Alismail S., Zhang H.and Chatterjee S. (2017) A Framework for Identifying Design Science Research Objectives for Building and Evaluating IT Artifacts. In: DESRIST 2017: Designing the Digital Transformation. Lecture Notes in Computer Science. Springer, pp. 218–230

Amarsy, N. (Oct. 15, 2015) How to scan your business model environment for disruptive threats and opportunities

Avdiji H., Elikan D., Missonier S., Pigneur Y. (2020) A Design Theory for Visual Inquiry Tools. In: Journal of the Association for Information Systems 21(3), pp. 695–734

Babar Z., Yu E. (2019) Digital Transformation – Implications for Enterprise Modeling and Analysis. In: 2019 IEEE 23rd International Enterprise Distributed Object Computing Workshop Proceedings. IEEE, pp. 1–8

Baskerville R., Pries-Heje J. (2010) Explanatory Design Theory. In: Business & Information Systems Engineering 2(5), pp. 271–282

Becker J., Probandt W., Vering O. (2012) Grundsätze ordnungsmäßiger Modellierung. Springer

Bergmann A., Strecker S. (2018) A Modeling Method in Support of Strategic Planning: Language Design and Application. In: Proceedings of the International Conference on Information Systems

Bider I., Perjons E., Bork D. (2021a) Context is King: an Enterprise Model that Connects the Internal Structure with the Business Environment. In: Proceedings of the IEEE International Enterprise Distributed Object Computing Workshop. IEEE, pp. 333–336

Bider I., Perjons E., Bork D. (2021b) Towards On-The-Fly Creation of Modeling Language Jargons. In: Proceedings of the International Conference on ICT in Education, Research, and Industrial Applications. CEUR-WS.org

Bork D., Alter S. (2018) Relaxing Modeling Criteria to Produce Genuinely Flexible, Controllable, and Usable Enterprise Modeling Methods. In: Proceedings of the International Workshop on Enterprise Modeling and Information Systems Architectures. CEUR-WS.org

Bork D., Alter S. (2020) Satisfying Four Requirements for More Flexible Modeling Methods: Theory and Test Case. In: Enterprise Modelling And Information Systems Architectures Journal 15, pp. 1–25

Bork D., Buchmann R., Hawryszkiewycz I., Karagiannis D., Tantouris N., Walch M. (2016) Using Conceptual Modeling to Support Innovation Challenges in Smart Cities. In: Proceedings of the International Conference on High Performance Computing and Communications; International Conference on Smart City; International Conference on Data Science and Systems. IEEE, pp. 1317–1324

Brinkkemper S. (1996) Method Engineering - Engineering of Information Systems Development Methods and Tools. In: Information and Software Technology 38(4), pp. 275–280

Buckl S., Matthes F., Monahov I., Schweda C. M. (2011) Modeling Enterprise Architecture Transformations. Working paper. Technical University of Munich, Chair for Software Engineering for Business Information Systems

Ciriello R. F., Richter A., Schwabe G. (2018) The paradoxical effects of digital artefacts on innovation practices. In: European Journal of Information Systems 28(2), pp. 149–172

Colbert, B. (June 15, 2021) Enterprise Architecture Inside Salesforce

de Kinderen S. (2017) Model Bundling: Componential Language Engineering. In: Architectural Coordination of Enterprise Transformation. The Enterprise Engineering Series. Springer, pp. 221–233

de Kinderen S., Kaczmarek-Heß M., Ma Q., Razo-Zapata I. S. (2021) A Modeling Method in Support of Strategic Analysis in the Realm of Enterprise Modeling. In: Enterprise Modelling And Information Systems Architectures Journal 16, pp. 1–36

de Kinderen S., Ma Q. (2015) Requirements Engineering for the Design of Conceptual Modeling Languages: A Goal- and Value-Oriented Approach. In: Applied Ontology 10(1), pp. 7–24

de Kinderen S., Proper H. A. (2013) e3-RoME: a value-based approach for method bundling. In: Proceedings of the Symposium On Applied Computing. ACM, pp. 1469–1471

Demirkan H., Spohrer J. C., Welser J. J. (2016) Digital Innovation and Strategic Transformation. In: IT Professional 18(6), pp. 14–18

Drechsler A., Hevner A. R. (2018) Utilizing, Producing, and Contributing Design Knowledge in DSR Projects. In: DESRIST 2018: Designing for a Digital and Globalized World. Lecture Notes in Computer Science. Springer, pp. 82–97

Fill H.-G. (2020) Enterprise Modeling: From Digital Transformation to Digital Ubiquity. In: Proceedings of the Federated Conference on Computer Science and Information Systems. IEEE, pp. 1–4

Fließ S., Kleinaltenkamp M. (2004) Blueprinting the service company. In: Journal of Business Research 57(4), pp. 392–404

Frank U., van Laak B. (2003) Anforderungen an Sprachen zur Modellierung von Geschäftsprozessen. Work report. Institut für Wirtschaftsinformatik Universität Koblenz-Landau

Gutzwiller T. (1994) Das CC RIM-Referenzmodell für den Entwurf von betrieblichen, transaktionsorientierten Informationssystemen. Physica

Hafsi M., Assar S. (2021) Proposal of a visual impact analysis approach based on enterprise modeling: case of customer experience projects in the banking sector. In: Proceedings of the PoEM 2021 Workshops. CEUR-WS.org

Hevner A. R. (2007) A Three Cycle View of Design Science Research. In: Scandinavian Journal of Information Systems 19(2), pp. 87–92

IMG (1997) PROMET BPR, Methodenhandbuch für den Entwurf von Geschäftsprozessen Version 2.0

Jöhnk J., Ollig P., Rövekamp P., Oesterle S. (2022) Managing the complexity of digital transformation — How multiple concurrent initiatives foster hybrid ambidexterity. In: Electronic Markets 32(2)

Kaplan R. S., Norton D. P. (1992) The Balanced Scorecard — Measures that Drive Performance. In: Harvard Business Review 70(1), pp. 71–79

Karagiannis D., Kühn H. (2002) Metamodelling Platforms. In: EC Web 2002: E-Commerce and Web Technologies. Lecture Notes in Computer Science. Springer

Kautz T. (2022) Digital Innovation Designer: Ein Business Engineering basiertes Tool zum systematischen Schliessen der Lücke zwischen Ideation und erfolgreicher Implementierung von Geschäftsinnovationen. Unpublished master thesis, University of St.Gallen

Kautz T., Winter R. (in press) Towards Tailored Support For Design Practices In Collaborative Modeling Sessions

Koutsopoulos G., Andersson A., Stirna J., Henkel M. (2022) Applying and Evaluating the KYKLOS Method. In: The Practice of Enterprise Modeling. Lecture Notes in Business Information Processing. Springer, pp. 118–133

Kurpjuweit S., Winter R. (2007) Viewpoint-based Meta Model Engineering. In: Proceedings of the International Workshop on Enterprise Modelling and Information Systems Architectures. Gesellschaft für Informatik, pp. 143–161

LeanIX (2022) Business Capability Map

Leist S. (2002) Bankenarchitektur des Informationszeitalters - Zielsetzung und Gestaltungsebenen In: Retail Banking im Informationszeitalter Springer, pp. 4–28

Levina N., Orlikowski W. J. (2009) Understanding Shifting Power Relations within and across Organizations: A Critical Genre Analysis. In: Academy of Management Journal 52(4), pp. 672–703

Lodhi A., Bider I. (2019) Using Enterprise Modeling in Development of New Business Models. In: Proceedings of the International Conference on Enterprise Information Systems. SciTePress, pp. 525–533

Maaravi Y., Heller B., Shoham Y., Mohar S., Deutsch B. (2021) Ideation in the digital age: literature review and integrative model for electronic brainstorming. In: Review of Managerial Science 15(6), pp. 1431–1464

McGinnis L. F. (2007) Enterprise modeling and enterprise transformation. In: Information, Knowledge, Systems Management 6(1-2), pp. 123–143

Miro (2022) Templates Library

Nicolini D., Monteiro P. (2016) The Practice Approach: For a Praxeology of Organisational and Management Studies. In: The SAGE Handbook of Process Organization Studies. SAGE, pp. 110–126

Object Management Group (2017) OMG® Unified Modeling Language® (OMG UML®)

Österle H., Winter R., Höning F., Kurpjuweit S., Osl P. (2007) Business Engineering: Core-Business-Metamodell. In: WISU - Das Wirtschaftsstudium 36(2), pp. 191–194

Osterwalder A. (2004) The business model ontology: A proposition in a design science approach. PhD thesis, University of Lausanne

Osterwalder A., Pigneur Y. (2010) Business model generation: A handbook for visionaries, game changers, and challengers. John Wiley & Sons

Osterwalder A., Pigneur Y., Bernarda G., Smith A. (2014) Value Proposition Design: How to Create Products and Services Customers Want. John Wiley & Sons

Oxford Advanced Learner's Dictionary (2022) Definition of collaborative adjective from the Oxford Advanced Learner's Dictionary

Pastor O., Noel R., Panach I., Ruiz M. (2022) The LiteStrat Modelling Method: Towards the Alignment of Strategy and Code. In: Domain-Specific Conceptual Modeling. Springer, pp. 141–159

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

Roelens B. (2022) PGA 2.0: A Modeling Technique for the Alignment of the Organizational Strategy and Processes. In: Domain-Specific Conceptual Modeling. Springer, pp. 121–139

Rummler G. A., Brache A. P. (2012) Improving Performance: How to Manage the White Space on the Organization Chart, 3rd ed. John Wiley & Sons

Sandkuhl K., Fill H.-G., Hoppenbrouwers S., Krogstie J., Matthes F., Opdahl A., Schwabe G., Uludag Ö., Winter R. (2018) From Expert Discipline to Common Practice: A Vision and Research Agenda for Extending the Reach of Enterprise Modeling. In: Business & Information Systems Engineering 60(1), pp. 69–80

Sandkuhl K., Stirna J., Persson A., Wißotzki M. (2014) Enterprise Modeling: Tackling Business Challenges with the 4EM Method. Springer

Schneider A. W., Zec M., Matthes F. (2014) Adopting Notions of Complexity for Enterprise Architecture Management. In: Proceedings of the Americas Conference on Information Systems

Silk E. M., Rechkemmer A. E., Daly S. R., Jablokow K. W., McKilligan S. (2021) Problem framing and cognitive style: Impacts on design ideation perceptions. In: Design Studies 74(C), pp. 1–30

Sonnenberg C., vom Brocke J. (2012) Evaluations in the Science of the Artificial: Reconsidering the Build-Evaluate Pattern in Design Science Research. In: DESRIST 2012: Advances in Theory and Practice. Lecture Notes in Computer Science. Springer, pp. 381–397

Stirna J., Persson A. (2012) Purpose Driven Competency Planning for Enterprise Modeling Projects. In: Advanced Information Systems Engineering. Lecture Notes in Computer Science. Springer, pp. 662–677

Teece D., Peteraf M., Leih S. (2016) Dynamic Capabilities and Organizational Agility: Risk, Uncertainty, and Strategy in the Innovation Economy. In: California Management Review 58(4), pp. 13–35

van den Oever M., Roelens B., Bork D. (2022) Practically Applicable Enterprise Models: A Research Project Toward a User-oriented Design Method. In: Proceedings of the International Workshop on Value Modelling and Business Ontologies. CEUR-WS.org

van Gils B., Proper H. A. (2018) Enterprise Modelling in the Age of Digital Transformation. In: The Practice of Enterprise Modeling. Lecture Notes in Business Information Processing. Springer, pp. 257–273

van Gils B., Weigand H. (2020) Towards Sustainable Digital Transformation. In: Proceedings of the Conference on Business Informatics. IEEE, pp. 104–113

Venable J. R., Pries-Heje J., Baskerville R. L. (2012) A Comprehensive Framework for Evaluation in Design Science Research. In: DESRIST 2012: Advances in Theory and Practice. Lecture Notes in Computer Science. Springer, pp. 423–438

Verhoef P. C., Broekhuizen T., Bart Y., Bhattacharya A., Dong J. Q., Fabian N., Haenlein M. (2021) Digital transformation: A multidisciplinary reflection and research agenda. In: Journal of Business Research 122, pp. 889–901

Vessey I., Galletta D. (1991) Cognitive Fit: An Empirical Study of Information Acquisition. In: Information Systems Research 2(1), pp. 63–84

Vogel O., Arnold I., Chughtai A., Ihler E., Kehrer T., Mehlig U., Zdun U. (2009) Software-Architektur. Spektrum Akademischer Verlag

Winter R. (2011) Business Engineering Navigator – Gestaltung und Analyse von Geschäftslösungen "Business-to-IT". Springer