M. Riebisch, M. Tropmann-Frick (Hrsg.): Modellierung 2022,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2022

Modeling the Enterprise Digital Twin: Towards an Open Platform for Analytics & Compliance Operations

Marc Frerichs[1], Markus Nüttgens[1]

Abstract: With its origins in the manufacturing and production sector, the concept of the Digital Twin has now become firmly established. Companies are increasingly relying on the potential advantages of operating a digital representation of their corporate structure with the Digital Twin of an Organization. This is done by incorporating relevant dynamic and operational data on the slower-changing structural data of traditional business. This paper deals with the question of how a digital twin can be designed and actually implemented, particularly taking into account data and processes extracted from business information systems. A real, implemented use case is used to show how operational decision-making can be supported by a digital twin and what role the concept of Hybrid Intelligence plays in this context. The findings of this work are the basis for an open platform for business analytics and resulting (semi-)automated actions, e.g., for implementation of compliance operations such as double payment analysis.

Keywords: Enterprise Digital Twin; Digital Twin of an Organization; Business Analytics

1 Introduction

According to Gartner, the concept of the Digital Twin (DT) is one of the most emerging innovations ("innovation profiles") [Da21]. DT is the most widely used innovation profile, showing the need for organizations to consider the impacts of their innovations. While DTs have been firmly established in the domain of manufacturing, companies are increasingly relying on the potential advantages of operating a digital representation of their corporate structure with the Digital Twin of an Organization (DTO). However, the creation of a DTO is a grand challenge [AHW21; Ca20], e.g., because the boundaries of an organization are not clear. In this paper, we propose the notion of the so-called Enterprise Digital Twin (EDT)—an approach for a generic/reference architecture to support the creation of DTOs in an enterprise context, i.e., especially aiming for organizations using business information systems such as Enterprise Resource Planning (ERP) systems. Furthermore, we conduct a case study to show feasibility of the approach by developing and using a software prototype. In this work, we address the following research questions:

RQ1: How can an EDT architecture be designed? Using Design Science Research (DSR), especially design principle induction, we derive a generic system architecture representing fundamental parts relevant to the realization of a DTO.

---

[1] Universität Hamburg, Hamburg Research Center for Information Systems (HARCIS), {marc.frerichs, markus.nuettgens}@uni-hamburg.de

doi:10.18420/modellierung2022-008

RQ2: How can a DTO based on the EDT architecture be implemented? By conducting a case study, including the development of a software prototype, we show how a DTO can be implemented on the basis of the EDT architecture.

The paper is structured as follows: First, the background and related work is discussed briefly. Afterwards, the methodology is described. The next section deals with the development of the EDT architecture. This is followed by a case study in the context of double payment analysis. The paper closes with a conclusion and outlook.

2 Background

DTs [Gr14] have emerged as a popular research field and also seem to raise high expectations in practice. A high level of relevance can be observed above all in the area of manufacturing, but also in aviation and health care [BCF19]. A DTO [KK21] aims to represent the elements and connections of an organizational system in virtual models in order to continuously simulate and analyze the behavior of the organization [Ca20]. *Process Mining* [Aa16] can help to facilitate such a DTO by providing methods to discover the control-flow model [PA21]. Van der Aalst et al. propose *Hybrid Intelligence* (HI)—"the ability to achieve complex goals by combining human and artificial intelligence" [De19b]—to make DTOs more resilient to unprecedented situations, and thus coined the term *Resilient Digital Twin* [AHW21]. To the best of our knowledge, existing approaches in the context of generic or reusable DT architectures primarily focus on industrial cases like Cyber-Physical Production Systems or Industry 4.0 [Bo21].

3 Methodology

The research presented follows the research approach of Design Science Research (DSR), which aims to develop artifacts that contribute to the solution of an existing problem [He04]. Österle et al. [Ös11] suggest a framework that divides DSR into the phases analysis, design, evaluation, and diffusion. Accordingly, our paper is structured corresponding to these phases (Fig. 1). For the design of the proposed artifacts, we used design principles induction [KGM12] to determine design principles (DPs) as the basis for an EDT architecture, and prototyping to develop a software artifact that implements the intended core functionality [NJ82]. DPs are generic, high-level representations derived from (meta-)requirements (MRs) [Ko16]. After analyzing existing knowledge regarding the realization of DTs in an enterprise context, we distill challenges and requirements. From these requirements, we derive DPs, which are the basis for a generic, high-level EDT architecture. For the implementation of a prototype, design decisions (DDs), i.e., instantiable and tangible features, are deduced. Several scholars highlight the importance of rigorous evaluation in design science-oriented research [RSB09; VPB12]. We therefore evaluate the proposed notion of an EDT by implementing a working prototype and validate it against a real-world scenario in a case study.

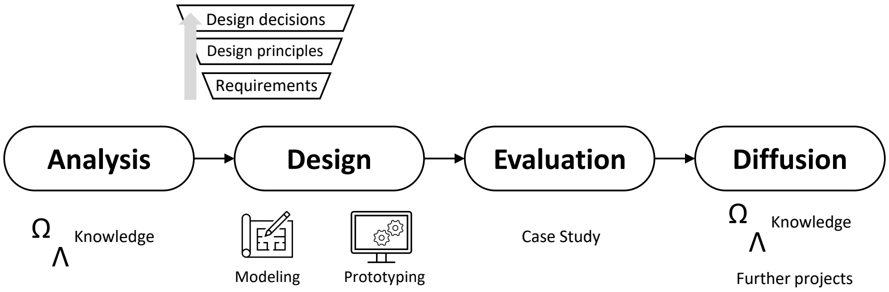


Fig. 1: DSR phases applied in this paper

4 Enterprise Digital Twin

Approaching a characterisation of DTs in general, different definitions, challenges, requirements, or properties can be named [Jo20; Se21]. Arguably the lowest common denominator in defining the concept of DT is that it is a digital representation of a real-world entity or system with bi-directional data connections [Gr14]. One area of application that has gained importance in recent years is the representation of an organization using DTs, also called DTO. Creating a DTO is one of the grand challenges in information systems [AHW21; Ca20]. With our work we want to narrow down the concept of a DTO a little and establish the notion of the EDT. The EDT provides a basic framework for realization of a DTO based on data extracted from enterprise information systems and intends to provide orientation when developing such DTOs. The approach is independent of a specific application scenario and instead provides a reference for the implementation of a concrete DTO. In this section, we first examine challenges and requirements, after which meta-requirements are derived. We then induce generic DPs and propose an architecture that corresponds to these principles. The EDT architecture and respective DPs are the basis for the deduction of concrete DDs.

4.1 Challenges and Requirements

Based on the definition by Caporuscio et al. [Ca20], essential aspects of a DTO can be inferred: DTOs should be able to *represent all elements and connections of an organization* in virtual models, which can be perpetually simulated and analyzed to achieve *continuous assessment* and *optimization of the organization*. Van der Aalst [Aa21a] derives an abstract definition of a DT by describing the differences between a digital model, a digital shadow, a DT, and a resilient DT. By introducing resilient DTs, Van der Aalst et al. [AHW21] complement the three definitions with a type of DT in which HI [De19a] comes into play. The complementary strengths of HI are presented in two scenarios [Aa21b; De19b]: (1) AI in the loop of human intelligence—support in decision-making by humans using AI, and (2) human intelligence in the loop of AI—humans continuously train machine learning models to improve the results of the AI. Becker and Pentland [BP22] state that organizations involve agents, typically human, with the ability to make choices and learn from experience. The bi-directional data connection between DT and reality is also referred to as twinning [Jo20]. To ensure twinning, continuous data updates are necessary. Furthermore, both historical and current data is needed as a basis of a DT [AHW21]. Thus, for the operation of an EDT, both initial (for historical data) and continuous data transfer (for current/real-time data) are required. With regard to data extraction, especially in terms of process mining, but in data-driven projects in general, scholars state that about 80% of the time is spent with ETL-related activities such as data access or data preparation [Sa21]. We therefore deduce that this problem must also be dominant in the context of DTs. When it comes to relevant data for an EDT, it must be specified how the state and behavior of information systems are represented in an EDT. According to Polyvyanyy and Van der Werf [PW20], data and processes in information systems go hand in hand. Becker and Pentland [BP22] state that organizations include multiple, interdependent processes. We deduce that both data—in business information systems one often speaks of documents, business objects or events—as well as different types of processes must be mapped in an EDT. As stated above, the EDT approach is "use-case agnostic". For example, Park and Van der Aalst [PA21] propose triggers based on constraint checking in their approach based on action-oriented process mining. While this is practical for this particular use case, it may not be for another purpose. We therefore expect an EDT to be open to (highly specialized) tools that are most suitable for the respective use case [FLN21]. Based on the previous statements, we derive key challenges and requirements to be considered when designing the EDT (Tab. 1).

Tab. 1: Key challenges and requirements

	Key challenge/requirement	References
(a)	Represent all elements and connections of an organization	[Ca20]
(b)	Continuous assessment and optimization of the organization	[Aa21a; AHW21; Ca20]
(c)	Twinning / bi-directional data connection	[Aa21a; Gr14; Jo20]
(d)	Data and processes as basic representation of the organization or underlying information system	[BP22; PW20]
(e)	Historical and current data	[AHW21; Ca20]
(f)	Combination of human and machine intelligence	[Aa21b; AHW21; BP22; De19a; De19b]
(g)	Tool-agnostic EDT	[FLN21; Ri05]

4.2 Design of the EDT

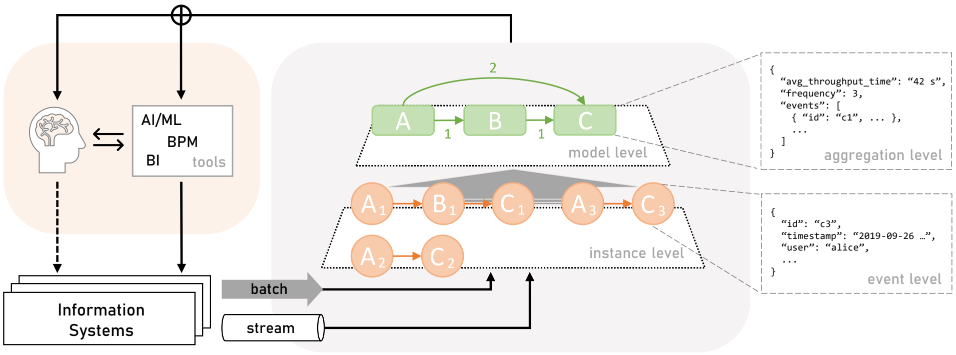
According to the methodology described in Sect. 3, the next step is to derive meta-requirements that aggregate the essential requirements of an EDT. Afterwards, design principles are induced, which are used for the design of an EDT architecture.

Meta-Requirements Aggregation. We cluster key challenges (a) and (d) as first meta-requirement (MR1) since they both address the representation of the organization in the EDT. (b), (c) and (e) affect (continuous) data transfer between "real" (in this case one or more information system(s)) and virtual entity (EDT), considering both historical and current data, thus leading to meta-requirement MR2. (f) addresses the involvement of humans in the EDT, significantly inspired by the concept of HI and resilient DTs, resulting in meta-requirement MR3. The idea of a (method- and) tool-agnostic EDT (g) is a result of the generic notion of the EDT, and forms the last meta-requirement (MR4).

Tab. 2: Design principles, meta-requirements and key challenges

	Design principle	Meta-requirement	Addresses key challenge(s)
DP1	Consider data and processes on different abstraction levels	*MR1* Adequate representation of the organization with data and processes	a, d
DP2	Use data exchange interfaces supporting batch and continuous procedure calls	*MR2* (Continuous) data transmission, considering both historical and current data	b, c, e
DP3	Enable 1) fully manual, 2) fully automated, and 3) hybrid decision-making	*MR3* Decision-making including Hybrid Intelligence	f
DP4	Provide adequate generic interfaces for tools to access business and process data	*MR4* Method- and tool-agnostic EDT	g

Design Principles Induction. In the following, the induction of DPs is described. They provide the basis for the design of the EDT architecture. Since the design of the EDT should consciously not be dependent on a specific use case, we propose a representation of the processes on different levels of abstraction, i.e. instance and model level [KK02; We12]; addressing MR1. On the one hand, one would want to investigate single executions of a case, e.g., for purposes of compliance checking [We17], while on the other hand an aggregated ("managerial") level of abstraction, e.g., for calculation of performance measures, is needed [DA05]. Consequently, this leads to design principle DP1, which aims for consideration of data and processes on different abstraction levels. To fulfill meta-requirement MR2, we propose an EDT with interfaces to the respective information systems (in and out), capable of both batch (for historical data) and continuous data transmission (DP2). Meta-requirement MR3 deals with HI to involve humans, either as part of decision-making (AI in the loop of human intelligence) or to train AI (human in the loop of AI). However, since a DT must still be able to make autonomous decisions [Aa21a], there should be several options for decision-making or "intervention in reality". We therefore propose 1) fully automated decision-making, 2) fully manual decision-making by humans, and 3) decision-making using HI, thus defining DP3. Next to meta-requirement MR2, the idea of a method- and tool-agnostic EDT, as described in meta-requirement MR4, is still more of a challenge than a requirement. A method- and tool-agnostic EDT should at least provide reasonable generic interfaces so the respective tools can benefit from the data and processes available in the EDT (DP4). However, the continuous development of a number of special interfaces tailored to the respective tools would be better. For example, this would be conceivable in an open source setting. Tab. 2 shows the derived DPs, the underlying meta-requirements, and key challenges.



```
{
  "avg_throughput_time": "42 s",
  "frequency": 3,
  "events": [
    { "id": "c1", ... },
    ...
  ]
}
```

```
{
  "id": "c3",
  "timestamp": "2019-09-26 …",
  "user": "alice",
  ...
}
```

Fig. 2: Enterprise Digital Twin

Modeling the EDT. Based on the DPs, we create a generic, high-level architecture of the EDT. It essentially consists of three parts: (1) the EDT itself, (2) a human-in-the-loop and tool part, and (3) the information system(s) representing the organization. As DP1 prescribes, the EDT should contain different abstraction levels so that different needs are addressed. Process instances are stored on an instance level, with events containing detailed information, e.g., a document containing a document ID or name of the responsible user. These process instances can be aggregated (model level), i.e., a process model is created, with activities containing aggregated data such as frequency or throughput time. The EDT is fed with data from the information system(s). This is possible via batch ("one-time shot" with potentially high data volume) and/or event stream (data is produced and transferred continually). In Fig. 2, this is indicated by two parallel processing pipelines (DP2). The "way back" to the information systems is accomplished via tool integration (DP3). DP3 and DP4 are realized via what we called human-in-the-loop and tool part. As described in Sect. 4.1, decision-making (and the like) should not be integrated into EDT, but outsourced to tools. This prevents us from reinventing the wheel (just for the sake of integrating it into the EDT) and leaves this job to tools that are more suitable. Since humans can also be involved in decision-making, the human-in-the-loop was integrated in the architecture. In this case, the human interacts with the tool(s) such as an AI or a business intelligence tool. Furthermore, the architecture also allows only the human (i.e., fully manual; e.g., by taking an action via the user interface of an information system) and only the machine/tool (i.e., fully automated; e.g., via a system interface such as BAPIs[2] in SAP) to make decisions.

---

[2] BAPI: Business Application Programming Interface; a standardized API for SAP business objects

Tab. 3: Design decisions derived from design principles

		DP(s)
DD1	Property graph as target data structure for process instance graphs using a graph database	DP1
DD2	Instance aggregation algorithm for construction of process models	DP1
DD3	Data extraction component allowing batch and continuous extraction of accounting data from an ERP system	DP2
DD4	Bulk importer for initial batch imports; message queue for processing data streams	DP2
DD5	REST interface for accessing business and process data	DP3, DP4

5 Evaluation

The goal of the case study is the concrete application of the design principles and the resulting reference architecture for EDTs as a basis for the development of a software artifact. The prototype created is used to analyze double payments in companies. A double payment is the case when a company pays the full equivalent more than once for a delivery or service [BRW11]. Obviously, double payments pose risks for businesses as they increase costs and reduce profits. The prototypical implementation is based on concrete, implementable DDs derived from the DPs (see Tab. 3).

With SAP S/4HANA, we chose a widely used ERP system as “object of investigation”. For data extraction, a Java-based component was developed, supported by the SAP Java Connector[3] (DD3), which provides an API that enables Java applications to communicate with SAP systems via SAP’s Remote Function Call (RFC) protocol. For continuous data streaming, Apache Kafka[4] (and Confluent[5]) is used (DD4). Neo4j[6] serves as a graph database for the persistence of the process instance graphs (DD1). Batch import of historical data is accomplished via the Neo4j Bulk Importer[7] (DD4). Aggregation of instances to process models is performed by a Java-based tool that can generate directly-follows graphs (DD2) and provides a REST API (DD5) for business data (document/event level) and process data (event log).

For double payment analysis, mostly journal entry documents are needed. In SAP, journal entries are stored in table BKPF; journal entry items can be found in table BSEG. Fig. 3 shows the data extraction part of the prototypical implementation described above, i.e., the path from the SAP source system to the process instances in Neo4j.

---

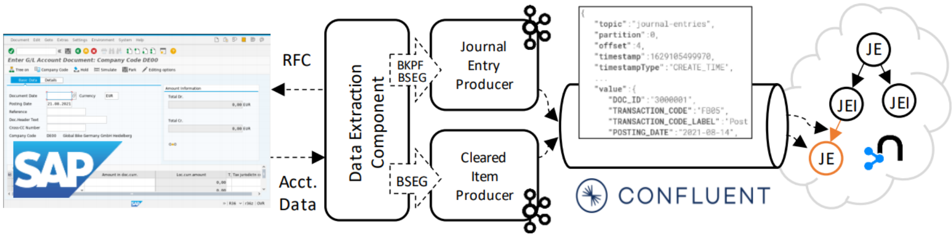
[3] `https://support.sap.com/en/product/connectors/jco.html`

[4] `https://kafka.apache.org/documentation/`

[5] `https://docs.confluent.io/platform/current/quickstart/ce-docker-quickstart.html`

[6] `https://neo4j.com`, `https://github.com/neo4j-contrib/neo4j-streams`

[7] `https://neo4j.com/developer/guide-import-csv/#batch-importer`



```
{
   "topic":"journal-entries",
   "partition":0,
   "offset":4,
   "timestamp":1629105499970,
   "timestampType":"CREATE_TIME",
   ...
   "value":{
      "DOC_ID":"3000001",
      "TRANSACTION_CODE":"FB05",
      "TRANSACTION_CODE_LABEL":"Post
      "POSTING_DATE":"2021-08-14",
```

Fig. 3: Part of the prototypical implementation

Our EDT-based application supports the user in detecting double payments. An AI-based tool (an external tool in the sense of the EDT) suggests potential double payments to the user. This is done in form of a list of double payment candidates, each associated with a probability of being a double payment. Based on their (business) experience, the user can assess and judge whether it is a double payment. By rating a double payment case, the user not only trains the AI and thus continuously improves it, but also manages the double payment cases. Since this involves interventions in a critical business system, a reclaim from the supplier or the cancellation of completed double payments does not take place automatically, but through manual actions by the user. The interaction of the user with the AI (training/judgement) and the decision-making support by the AI are good examples of human-in-the-loop. By rating individual candidates, the AI model adapts and with it the probability of the cases. The case study showed that an implementation based on the EDT architecture and the underlying design principles is feasible. Furthermore, a concrete application of the developed prototype was shown.

6 Conclusion and Outlook

In this research paper we showed how DTs can be realized based on a generic EDT architecture. Based on the initial requirements, design principles were determined using design principles induction, which served as the basis for the EDT architecture (RQ1). Using the proposed architecture and the design decisions derived from the DPs, we created a prototype and implemented a real use case (RQ2). Obviously this paper is not without limitations and provides a first step towards a generic and high-level, tool-agnostic EDT. One of the probably biggest challenges was already mentioned in Sect. 4.2: the data exchange with information systems is difficult to generalize, since each system has different, often proprietary interfaces [FLN21]. The same applies to the interfaces of the tools. This is where the OpenPACO (Open Platform for Analytics & Compliance Operations) project comes into play. It aims to close the gap between business information systems such as ERP systems and analytics or modeling tools. The goal is an open-source platform that can be constantly extended with new interfaces to source and target systems. This paper forms a basis for further activities in this context.

References

[Aa16] van der Aalst, W. M. P.: Process Mining. Springer Berlin Heidelberg, 2016.

[Aa21a] van der Aalst, W. M. P.: Concurrency and Objects Matter! Disentangling the Fabric of Real Operational Processes to Create Digital Twins. In: Theoretical Aspects of Computing – ICTAC 2021. Pp. 3–17, Aug. 2021.

[Aa21b] van der Aalst, W. M. P.: Hybrid Intelligence : to automate or not to automate, that is the question. International Journal of Information Systems and Project Management 9/, pp. 5–20, 2021.

[AHW21] van der Aalst, W. M. P.; Hinz, O.; Weinhardt, C.: Resilient Digital Twins. Business & Information Systems Engineering 63/, pp. 615–619, Dec. 2021.

[BCF19] Barricelli, B. R.; Casiraghi, E.; Fogli, D.: A Survey on Digital Twin: Definitions, Characteristics, Applications, and Design Implications. IEEE Access 7/, pp. 167653–167671, 2019.

[Bo21] Bolender, T.; Bürvenich, G.; Dalibor, M.; Rumpe, B.; Wortmann, A.: Self-Adaptive Manufacturing with Digital Twins. In: 2021 International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS). Pp. 156–166, 2021.

[BP22] Becker, M. C.; Pentland, B. T.: Digital Twin of an Organization: Are You Serious? In: Business Process Management Workshops. Springer International Publishing, Cham, pp. 243–254, 2022.

[BRW11] Bönner, A.; Riedl, M.; Wenig, S.: Doppelzahlungen. In: Digitale SAP-Massendatenanalyse. ESV Erich Schmidt Verlag, 2011.

[Ca20] Caporuscio, M.; Edrisi, F.; Hallberg, M.; Johannesson, A.; Kopf, C.; Perez-Palacin, D.: Architectural Concerns for Digital Twin of the Organization. In: Software Architecture. Springer International Publishing, Cham, pp. 265–280, 2020.

[DA05] van Dongen, B.; van der Aalst, W. M. P.: Multi-phase process mining: Aggregating instance graphs into EPCs and Petri nets. Proceedings of the Second International Workshop on Applications of Petri Nets to Coordination, Workflow and Business Process Management/, 2005.

[Da21] Dawson, P.: 2021 Hype Cycles: Innovating Delivery Through Trust, Growth and Change. Gartner/, 2021.

[De19a] Dellermann, D.; Calma, A.; Lipusch, N.; Weber, T.; Weigel, S.; Ebel, P.: The Future of Human-AI Collaboration: A Taxonomy of Design Knowledge for Hybrid Intelligence Systems. In: 52nd Hawaii International Conference on System Sciences (HICSS). Jan. 2019.

[De19b] Dellermann, D.; Ebel, P.; Söllner, M.; Leimeister, J. M.: Hybrid Intelligence. Business & Information Systems Engineering 61/, pp. 637–643, Oct. 2019.

[FLN21] Frerichs, M.; Leible, S.; Nüttgens, M.: Towards method- and tool-independent business process modeling. In: INFORMATIK 2021, Lecture Notes in Informatics (LNI). Gesellschaft für Informatik (GI), Bonn, 2021.

[Gr14] Grieves, M.: Digital Twin: Manufacturing Excellence through Virtual Factory Replication, 2014.

[He04] Hevner, A. R.; March, S. T.; Park, J.; Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28/, 2004.

[Jo20] Jones, D.; Snider, C.; Nassehi, A.; Yon, J.; Hicks, B.: Characterising the Digital Twin: A systematic literature review. CIRP Journal of Manufacturing Science and Technology 29/, pp. 36–52, 2020.

[KGM12] Koppenhagen, N.; Gaß, O.; Müller, B.: Design Science Research in Action - Anatomy of Success Critical Activities for Rigor and Relevance. 20th European Conference on Information Systems (ECIS 2012)/, 2012.

[KK02] Karagiannis, D.; Kühn, H.: Metamodelling Platforms. In: E-Commerce and Web Technologies. Springer, Berlin, Heidelberg, pp. 182–196, 2002.

[KK21] Kerremans, M.; Kopcho, J.: Create a Digital Twin of Your Organization to Optimize Your Digital Transformation Program. Gartner Research/, 2021.

[Ko16] Koppenhagen, N.; Mueller, B.; Maedche, A.; Li, Y.; Hiller, S.: Designing a supply network artifact for data, process, and people integration. Information Systems and e-Business Management 14/, Aug. 2016.

[NJ82] Naumann, J. D.; Jenkins, A. M.: Prototyping: The New Paradigm for Systems Development. MIS Quarterly 6/, 1982.

[Ös11] Österle, H.; Becker, J.; Frank, U.; Hess, T.; Karagiannis, D.; Krcmar, H.; Loos, P.; Mertens, P.; Oberweis, A.; Sinz, E. J.: Memorandum on design-oriented information systems research. European Journal of Information Systems 20/, pp. 7–10, 2011.

[PA21] Park, G.; van der Aalst, W. M. P.: Realizing A Digital Twin of An Organization Using Action-oriented Process Mining. In: 2021 3rd International Conference on Process Mining (ICPM). Pp. 104–111, 2021.

[PW20] Polyvyanyy, A.; van der Werf, J. M. E. M.: Information Systems Modeling: Playing with the Interplay Between Data and Processes. In: 18th International Conference, BPM 2020. 2020.

[Ri05] Riggio, R.; Ursino, D.; Kühn, H.; Karagiannis, D.: Interoperability in meta-environments: An XMI-based approach. In: Advanced Information Systems Engineering. CAiSE 2005. Springer, Berlin, Heidelberg, pp. 77–89, 2005.

[RSB09] Riege, C.; Saat, J.; Bucher, T.: Systematisierung von Evaluationsmethoden in der gestaltungsorientierten Wirtschaftsinformatik. In: Wissenschaftstheorie und gestaltungsorientierte Wirtschaftsinformatik. Physica-Verlag HD, Heidelberg, pp. 69–86, 2009.

[Sa21] Sadiq, S. W.; Someh, I. A.; Chen, T.; Indulska, M.: Value Creation from Data - Why is this a BPM Problem? In: Proceedings of the International Workshop on BPM Problems to Solve Before We Die (PROBLEMS 2021) co-located with the 19th International Conference on Business Process Management (BPM 2021), Rome, Italy, September 6-10, 2021. Vol. 2938. CEUR Workshop Proceedings, CEUR-WS.org, pp. 11–17, 2021.

[Se21] Semeraro, C.; Lezoche, M.; Panetto, H.; Dassisti, M.: Digital twin paradigm: A systematic literature review. Computers in Industry 130/, p. 103469, 2021.

[VPB12] Venable, J.; Pries-Heje, J.; Baskerville, R.: A Comprehensive Framework for Evaluation in Design Science Research. In: Design Science Research in Information Systems. Advances in Theory and Practice. Springer, pp. 423–438, 2012.

[We12] Weske, M.: Business Process Modelling Foundation. In: Business Process Management: Concepts, Languages, Architectures. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 73–124, 2012.

[We17] Werner, M.: Financial process mining - Accounting data structure dependent control flow inference. International Journal of Accounting Information Systems 25/, pp. 57–80, 2017.