

Behavior-Centered Digital-Twin Design for Dynamic Cyber-Physical System Development

Christian Stary^{1*}, Matthes Elstermann², Albert Fleischmann³, and Werner Schmidt⁴

¹Business Informatics - Communications Engineering, Johannes Kepler University Linz, Linz, Austria

²Institute for Information Management in Engineering, Karlsruhe Institute of Technology, Karlsruhe, Germany

³InterAktiv Unternehmensberatung, Pfaffenhofen, Germany

⁴Business School, Technische Hochschule Ingolstadt, Ingolstadt, Germany

Christian.Stary@jku.at, matthes.elstermann@kit.edu,
albert.fleischmann@interaktiv.expert, werner.schmidt@thi.de

Abstract. Digital Twins are digital models of Cyber-Physical Systems to enable not only continuous monitoring but also active functional improvement of networked services, physical products, machines and devices. This capacity is of utmost importance when recognizing and exploring business opportunities in terms of organizational and technology innovations, as well as enriching the scope of system-relevant applications. Before being operated in their target ecosystems, such as smart cities, Cyber-Physical Systems can be validated and be run as Digital Twin through executable behavior models. The development of these models captures both, the horizontal, and the vertical integration of CPS components, thus allowing to consider specific system qualities, such as pollution effects of traffic. This article investigates methodological and technological aspects of developing and operating Digital Twins along system transformation processes. We consider integration depth and breadth, connectivity, organizational intelligence, validation, and implementation variability in the context of human-centered modeling and development. The approach enriches the understanding of digital twins towards digital representation of Cyber-Physical Systems allowing for dynamic allocation of physical and digital parts according to operational conditions. An exemplary case study in traffic management demonstrates the feasibility and practicability of the communication-centered approach.

Keywords: Digital Twin, Cyber-Physical System, Modeling, Subject Orientation, Design-Integrated Engineering.

1 Introduction

Digital transformation processes towards Cyber-Physical Systems (CPSs) have significant impact on ecosystems and organizations (cf. [1], [2]). Consider a traffic management system that can

* Corresponding author

© 2022 Christian Stary, Matthes Elstermann, Albert Fleischmann, and Werner Schmidt. This is an open access article licensed under the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>).

Reference: Ch. Stary, M. Elstermann, A. Fleischmann, and W. Schmidt, "Behavior-Centered Digital-Twin Design for Dynamic Cyber-Physical System Development," *Complex Systems Informatics and Modeling Quarterly*, CSIMQ, no. 30, pp. 31–52, 2022. Available: <https://doi.org/10.7250/csimq.2022-30.02>

Additional information. Author ORCID iD: Ch. Stary – <https://orcid.org/0000-0001-9764-5021> and W. Schmidt – <https://orcid.org/0000-0002-3088-8253>. PII S225599222200169X. Received: 31 October 2022. Accepted: 14 April 2022. Available online: 30 April 2022.

be successively enriched with intelligence to become part of an environment control system to include climate-specific factors in its behavior. Digitalization not only transforms the business and customer operations, but also development processes and the way stakeholders communicate with each other (cf. [3], [4]). Hence, digital transformation and managing its processes concerns a variety of constitutive business fields and resulting complex tasks, ranging from business models to operation, and requiring to rethink modeling practices in terms of evaluation and dynamic adaptation (cf. [5], [6], [7]).

Digital twinning has become an effective means to create a computational model of an application, to provide device shadows, mirror entire systems, or to engineer a synchronized virtual prototype [5]. Digital Twin (DT) technologies allow addressing complex domains such as home healthcare and logistics, as well as their intertwining with other complex domains, such as smart cities (cf. [8]). Thereby, DTs play themselves a transformative role 'not only in how we design and operate cyber-physical intelligent systems, but also in how we advance the modularity of multi-disciplinary systems to tackle fundamental barriers not addressed by the current, evolutionary modeling practices' [5].

When DTs are to be actually the living model of the physical system which continuously adapts to operational changes based on the real-time data from various IoT sensors and devices and forecasts the future of the corresponding physical counterparts with intelligence, they need to be human understandable and manageable in an interactive way. This requirement holds for modeling, abstractions, and inferences representing the intelligence of a CPS [9]. CPS development and operation involves a variety of stakeholders, not only in terms of technology but rather concerning the organization of tasks, and the type of business logic. For instance, for smart city operation traffic organizers, emergency services, facility and road management, drivers, and many other stakeholders are involved, pursuing a variety of tasks and utilizing various services and products. This involvement lays ground to a socio-technical system understanding of CPS with the DT as the baseline for development and operation. Stakeholders need to rely on their proper semantic understanding and DT models when handling structural information and operational dynamics. Human involvement may even require redundant forms of (re-)presentations (cf. [10]).

Hence, the objective of this research is to create a digital twin model of a CPS in a human-centered way while capturing the main characteristic for development and operation in a human-centered way. We target DT design and engineering through the horizontal and vertical integration of CPS components and their behavior models. Thereby, a behavior model of a CPS and its components is a representation encapsulating the execution of human or machine or combined activities to accomplish specific tasks. We are in line with understanding a digital twin model as 'an appropriately synchronized body of useful information (structure, function, and behaviour) of a physical entity in virtual space, with flows of information that enable convergence between the physical and virtual states. ... digital twins represent real objects or subjects with their data, functions, and communication capabilities in the digital world.' [5]

Since the behavior-modeling approach of Subject-oriented Business Process Management (S-BPM) allows meeting the resulting requirements with respect to methodological and technological support (cf. [11]), we utilize its capabilities and challenge them for Digital-Twin-based CPS development and operation. The approach targets the interaction and communication between active entities in the sense of acting parties in a networked environment and

- facilitates human understanding of models and specification processes due to the minimal set of intuitive notation elements – performing actions ('doing'), sending and receiving messages are connected by temporal relations
- targets executable specifications (DT models) to validate digital representations

Both features allow humans not only to design CPS using unifying behavior encapsulations (abstracting from the heterogeneity of CPS components regardless of being human or artificial), but

also to validate and run the resulting models interactively; and thus to engineer a CPS. The design can be probed and experienced by operating the DT models before putting the CPS to practice. Since the DT models make the CPS run-time behavior transparent for users, they can adapt the internal behavior of components when novel requirements occur in the course of operation.

The article is structured as follows: In Section 2, we detail the requirements for CPS development addressing the DT research problem, and detail the Design Science approach to develop a solution based on the kernel theory of S-BPM. In Section 3, we provide relevant background on DT development by structuring the DT dimensions according to fundamental capabilities required for behavior specification. They concern integration depth and breadth, connectivity of components for intelligent system behavior, and validation for operation and adaptation. In Section 4, we introduce subject-oriented design elements and the corresponding S-BPM-based development cycle of DTs. We use an exemplary case study from traffic management to demonstrate the feasibility and practicability of the approach. Section 5 concludes the article with reflecting on achievements and topic identification for future research.

2 Problem Statement and Methodological Approach

In a CPS and its development process various components and sub systems are represented in different ways, and respectively handled through different methods, including sensors, social media, and robot actuators. There it becomes a need to converge the different components in a model (cf. [12]) or to mutually align dedicated component models (cf. [13]). In that context, it also becomes imperative to help designers and users arrange components and information exchange in the way a CPS evolves.

DTs help in both ways. They not only facilitate decision making in the course of design through transparent modeling of behavior, but also allow to consider the contextual factors in terms of simulation, analytic processing and information exchange (cf. [14], [15]). Correspondingly, DTs are created from the convergence of CPS technologies and user behavior involving data exchange and communication.

Communication and data exchanges trigger behavior, i.e., sequences of activities performed by digital or physical components or actors to accomplish a task. Behavior models can capture single digital or physical CPS elements, such as a sensor or measurement task, as well as accumulated sets of activities, such as regulating living conditions of a person in home healthcare (cf. [16]).

However, each behavior representation needs to be understood in its context, as it influences the design of a DT and the engineering of the concerned CPS. Although DTs form the baseline of evolving or pro-active CPS behavior (cf. [17], [18]), the internal structure of DTs needs to be detailed to capture future functionalities and behavioral impact. These include the communication of DTs with their physical counterparts. Both, for design and operation, mutual communication is considered essential [19].

These requirements form the objective of our research, namely to support CPS developers, domain or technical experts, and users on creating and adapting digital twin models of a CPS in a human-centered way, while capturing the main characteristics of CPS. The latter mainly concern the connectivity and heterogeneity of components besides their distribution. Consequently, we target DT design and engineering through the horizontal and vertical integration of CPS components and their behavior models.

We consider the human-centered design and engineering requirement as being met once (i) a notation for specifying the digital model can be used by stakeholders with minimal diagrammatic skills and conceptual background, and (ii) the notation makes the run-time behavior of a CPS in a way transparent so that the digital twin model can be adapted in synchronization with the current CPS in operation.

The methodology to develop the human-centered integrated design and engineering support is solution-oriented and originates from Design Science [20], [21]. Its dual while iterative nature

with respect to design artifacts (models, software, services, machinery etc.) and theory recognition enables practical development and theory advancement. The core set of activities, namely development cycles focusing on an artifact representing the (part of the current) solution, connects to the situation and environment in terms of a relevant problem and its specific requirements for a solution. Each design cycle leads to results that become part of a knowledge base informing further problem solving steps.

In our research on CPS development support we follow Peffer's et al. operational framework [22] with the underlying kernel theory of communicating actors and subject-oriented modeling/execution. Each design cycle corresponds to a design or modeling step closer to engineering and finally, model execution. In this article the results of 3 successive design cycles are presented. Starting with (i) a DT specification of CPS components, we (ii) extended the behavior specifications with specific or exceptional situation information, before (iii) capturing intelligent behavior, including complex events. The demonstrator is a traffic management systems that evolves successively towards an environment management system by using a System-of-Systems perspective on the represented behaviors.

3 Digital Twins

In this section we provide conceptual background information on structuring DT approaches (Section 3.1) and continue giving dimensions according to their fundamental capabilities in relation to emergent and (pro-)active behavior specification (Section 3.2). Ranging from integration depth and breadth to execution capabilities when adapting to changes, we lay ground for structuring the discussion on dynamically adaptive behavior models introduced in the subsequent section. In Section 3.3 we review related work with respect to stakeholder-oriented design and engineering.

3.1 Concept

The understanding of the DT concept and its functioning is still under discussion (cf. [23]–[24]). There is consensus about the origin and fundamental nature of DTs as digital substitutes of some real world objects and their mutual connectedness. Originally defined as a virtual representation of a physical product containing information about that product, a DT aims to support product life-cycle management. Its characteristic structural components included the physical product and its virtual representation, and the bi-directional connection between the physical and the virtual representation to exchange and processes information.

The authors of [25] referred to various development stages of DTs, starting with the elementary type of a DT: A DT prototype provides steps for producing specific elements. The DT instance provides in-depth information on which parts are required to produce a specific instance of an element, including names and specifications, and what is needed to actually run a production process in real-time, including the sensors to be connected with the element. Finally, the DT aggregate considers a set of DT instances cumulating information of the concerned elements. The virtual representation of the environment within which the physical element, e.g., device, exists, is termed DT environment.

The DT starts life as a DT prototype in the initial design phase, and is followed by the realization phase, with DT instances created for each generated element. They are accumulated as DT instances by the DT aggregate within the DT environment. The latter enables applying digital services including model execution, modeling, and evaluation. The DT instances/aggregates and environment are considered to persist beyond the retirement or disposal phase of the physical elements.

The benefit of shadowing or twinning physical and virtual objects was expected in increasing performance through model execution and optimization, e.g., to validate product or system properties and support design and engineering tasks (cf. [14], [26]). Moreover, once a DT remains

existent while its physical counterpart is disposed or is at the end of its life cycle, knowledge of developing and operating the twinned counterpart can be preserved and utilized for developing objects of this kind or a CPS containing this type of objects.

The idea to add intelligence to DTs and use them to make forecasts of behavior and failures in systems is still one of the major driving forces, in particular, when coupling digital models in almost real-time to their physical counterpart in cyber-physical settings (cf. [27]).

The DT concept of the digital thread has been designed to emulate the life cycle, actions, and operation of its real-world counterpart aside (cf. [28]). It can reflect on any element, object, product, or system that works as its physical twin, capturing the inherent varying and dynamic nature of the physical world by capturing users and operators, assets, machinery, goods, specific regulations, and environmental context. Thereby, sensor systems, and thus Internet-of-Things (IoT) components of CPS, play a crucial role, as they are physically connected to devices and enable obtaining data and emerging information.

Built on top of so-called digital shadows referring to an automatic while unidirectional connection to capture the status of the real element through and its virtual counterpart, DT threads are enriched shadows and enable information interchange between physical and virtual components in a bidirectional way. Hence, the DT has the ability to be an instance of its physical counterpart and manage the entire system digitally. Any change of the real element or its virtual counterpart is captured by the other part. Hence, DT threads enable leveraging the benefits of both the virtual and physical environments to the benefit of the entire system. Information can be captured, stored, evaluated, and analyzed in the context of a current product while developing a future behavior or version of the product or an entire new product.

The prospect of DT threads is referred to as not only modeling the current situation of a CPS but having the capability to explore further variants of behavior. Once such a novel variant is accepted by its stakeholders after simulating or optimizing the current system or component, it can be implemented by putting the respective models to operation.

3.2 Capturing Variability of Behavior

In this section we summarize conceptualizations on desirable and undesirable behaviors [25] as essential parameter of (pro-)active DTs. DTs should have the capability to recognize variants of behavior that include operation without hindrances and interruption, as well as behavior that should be avoided in specific cases and in any case, e.g., to prevent harm to people and/or systems. Models, when represented, capture emergent behavior from a static perspective. Hence, we can use them to differentiate variants of behavior. Grieves and Vickers [25] separate predicted from unpredicted behavior, with each of them split into further 2 categories, differentiating desirable and undesirable behavior. We summarize these options in the Table 1 – the citations in the respective cells stem from [25], p. 90, the terms in [27] have been added by the authors for the sake of intelligibility:

- Predictable and desirable behavior can be considered the Happy Path, i.e., when a system and its components shows the behavior it has been designed for. It is traditionally envisioned in the course of meeting explicated requirements in an ideal(ized) way. For instance, in crisis management, all actors are available when being contacted and cooperate to manage the crisis in an informed and coordinated way.
- Predictable and undesirable behavior includes all cases that can be anticipated at design time hindering the Happy Path. Undesirable behavior can be addressed through modeling diagnostic and preventive activities. In case when events trigger such behavior, mechanisms exist to act for eliminating the underlying problem. For instance, when a traffic jam occurs a system can come up with alternative routes for traffic participants.
- Unpredictable and desirable behavior addresses cases which are part of Happy Path considerations but cannot be predicted. They are either beyond pre-specified requirements or

context and cannot be designed in terms of specifying them completely like other happy paths. For instance, the particular sequence of applying rules for calculating alternative routes in traffic management leads to a side effect that has not been considered at design time, such as the alternative route is also the most climate effective.

- Unpredictable and undesirable behavior subsumes all cases that should be avoided for specific reasons. Typical instances are physical settings leading to overreactions, such as overheating machinery, or deadlocks and hindrances, such as blocked highways for emergency vehicles, causing further harm and loss of control if not removed.

Table 1. DT Behavior Types according to [25]

Behavior Categories	Predictable	Unpredictable
Desirable	'the desired behavior of our system'; intentional design and realization of our system'; 'in systems engineering terms, it is the requirements our system is designed to meet.	'contains pleasant surprises. While ignorance is never bliss, this category will only hurt our pride of not having understood our system well enough to predict these beneficial behaviors.'
Undesirable	'contains problems we know about and will do something about eliminating. P[redictable] U[ndesirable] problems are management issues. If we know about the P[redictable] U[ndesirable] problems and do nothing about them, that is engineering malpractice. This is the category that expensive lawsuits are made of.'	'holds the potential for serious, catastrophic problems. It is this category of emergent behavior that we need to focus on.'

In addition, Grieves and Vickers [25] recognize emergent forms of behaviors, as they occur in the course of operating a system. They include alternatives that emerge from experiences or additional event recognition. There might be a discrepancy between the physical occurrence and the digital counterpart, as they have a different kind of timeliness and permanency in both directions. In case the variability or resilience in one 'world' changes, the other might lack action and/or representational means. (Pro-)Active DTs should be able to cope with all behavior types.

3.3 Related Work

Model-based engineering enables DT generation and targets the explicit representation of integrating digital models into CPSs and to systematically construct CPSs [29], [30]. The various models help capturing (i) the software part of a CPS, (ii) the DT information system, and (iii) the integration of (i) and (ii). Utilizing existing diagrammatic approaches, such as UML, class diagrams variants have been developed to generate the DT information system. The respective process for model-driven development allows to synchronize data between the physical and the virtual CPS parts, and is composed of several steps, namely to specify the CPS architecture and the DT information system before integrating them (cf. (i)-(iii) above). Finally, both the CPS and the DT information system are generated. Hence, programming the connection between the software part the DT information system can be automated.

With respect to the context of use, and thus human-centered design of CPS, Sandkuhl et al. [31] have addressed the organizational and business aspects beyond technological ones, referring to the suitability of enterprise modeling and capability management. Both affect the design and run-time of DTs. Once DT modeling and model management captures multiple perspectives (views) on CPSs and their development, the integration of organizational aspects and alignment with technological ones can be facilitated. DT models need also to enable continuous adaptation to capture evolving stakeholder needs and changing operational conditions. Thereby, the communication between components and the exchange of data play a crucial role. In data-driven

CPS design and engineering, e.g., as proposed in [32], product design has shifted its focus from the analysis of physical to virtual models, enabled by DTs. They represent a product mode to collect and accumulate product-relevant data continuously. In this way, the entire life cycle of a product, including design, manufacturing, quality control, and adaptation can be supported. DTs finally serve to preserve all product data by dynamically sensing, storing and analyzing them throughout the product life cycle.

In the tradition of data-driven development and adaptation, Wu et al. [33] suggested five dimensions to be captured by any DT: the physical entity, the virtual entity, the services module, the DT data module, and the connection module. The physical entity concerns the addressed tangible components of the DT as a point of CPS reference. It captures a DT's operational process, and thus the (environmental) data collection and the functional output. The virtual entity contains models of the physical entity, including its structure, function, and environment. It can also control the physical entity which is mainly done based on inputs from other models or entities for optimization. The services module allows stakeholders to define and utilize business functions via the DT. This module conveys relevant user information when operating the CPS. The DT data module is required for the DT data handling – it receives, stores, and delivers DT data from and to other components or systems over the DT lifetime. Finally, the connection module administrates all links between the various entities for real-time interaction across the modules.

Reflecting on the recently proposed approaches for DT development and operation support, model-based engineering is considered essential to represent DTs and generate functional software, thus generate DT and integrate them in CPSs. At the center of design are data and their exchange for real-time coupling of physical and digital parts of CPSs. When connectivity and the business/user perspective become design issues, several DT dimensions help balancing organizational requirements, technological capabilities, and user needs. In the following we ground the introduced subject-oriented DT approach on the concept of model-based engineering for human-centered design and operation. The modeling notation is based on a small set of elements and concepts, as its focus is on functions and communication, however, enabling automated model execution. In contrast to other development support approaches, we start modeling the behavior of the CPS and thus create a digital CPS representation before identifying its physical part in the organization and technology implementation phases. Our approach implements the virtual entity model of CPS mentioned in the framework reviewed above [33], however, integrated with the connection module functionality.

4 Behavior-centered DTs for CPS Development

In this section we introduce subject-oriented modeling and execution, as well relevant CPS design elements and the corresponding development cycle of DTs as threads. We demonstrate subject-oriented capabilities with respect to integration in terms of depth and breadth of processes, connectivity of elements, representing adaptive behavior according to intelligence capabilities, and validation of dynamic changes.

4.1 Subject-oriented CPS Modeling

Subject-orientation originates from sentence construction using subject, predicate, and object in human language, with subjects referring to active entities. These perform activities that result in behavior and concerns objects of manipulation. Utilizing subject-oriented modeling and execution capabilities [11], [34], [35], systems or subjects are viewed as emerging from both the interaction between subjects and their specific behaviors encapsulated within the individual subjects. Subjects as systems can operate in parallel and exchange messages asynchronously or synchronously.

Subject-oriented behavior models represent services connecting IoT devices / CPS components via message exchange carrying data in business objects, allowing transparency with respect to

internal data collection and processing, and interfaces to other components in terms of receiving and transmitting data. A CPS operates as a set of autonomous, concurrent behaviors of distributed components. These components or subjects are behavioral roles assumed by some entity that is capable of performing actions. The entity can be a human, a piece of software, a machine (e.g., a robot), a device (e.g., a sensor), or any combination of these, such as intelligent sensor systems.

When subject-oriented concepts and development techniques are applied, subjects can execute local actions that do not involve interacting with other subjects (e.g., calculating a threshold value for intervention and triggering an information process), and communicative actions that are concerned with exchanging messages between subjects, i.e. sending and receiving messages.

Subjects are one of five core symbols used in specifying designs. Based on these symbols, two types of diagrams can be produced to conjointly represent a system: Subject Interaction Diagrams (SIDs) and Subject Behavior Diagrams (SBDs).

SIDs provide an integrated view of a CPS, comprising the CPS components involved and the messages they exchange. SBDs provide a local view of the behavior from the perspective of individual subjects. They include sequences of states representing local actions and communicative actions. The latter comprise sending and receiving messages. State transitions are represented as arrows, with labels indicating the outcome of the preceding state. Given these capabilities, system designs are characterized by (i) simple communication protocols (using SIDs for a process overview) and thus, (ii) standardized behavior structures (enabled by send-receive pairs between SBDs), which (iii) scale in terms of complexity and scope. Subject-oriented design allows meeting ad-hoc and domain-specific requirements. As validated behavior specifications can be executed without further model transformation, stakeholders can guide the implementation of specifications, representing domain-specific task flows, and make ad-hoc changes by replacing individual subject behavior specifications during runtime.

Due to the distributed nature and loose coupling of subject-oriented representations, the ultimate stage of scalability could be reached through dynamic and situation-sensitive formation of CPS. Subject-oriented modeling is facilitated through patterns that re-occurred in various applications (cf. [34]). Once a Subject Behavior Diagram is instantiated, it has to be decided (i) whether a human or a digital device (organizational implementation) and (ii) which actual device is assigned to the subject, acting as technical subject carrier (technical implementation). Typical subjects are smart devices, which can have Internet connectivity, including smart phones, tablets, laptops, healthcare devices, etc.

Subject-oriented runtime engines, such as the market-ready suites Metasonic (<http://metasonic.de>), actnconnect (<http://actnconnect.de>), and Compunity (<http://compunity.eu>), or the research prototypes Ueberflow [36] and SiSi [37] are computing infrastructures for model validation and execution. They provide low-latency virtualized services and can be linked with the Cloud Computing infrastructure by the same subject interaction mechanism. For instance, the open source engine UeberFlow [36], actions or tasks are ordered in the sequence as defined through SBDs and SIDs. The Workflow Specification of UeberFlow represents an entirely executable model of an application, given the subject actions and communication with others. It acts as container for so-called WorkflowUnits that are created for each subject, and captures all activities (WorkflowSteps). In addition, WorkflowUnit manages the data processed by the WorkflowSteps and its WorkflowFunctions. Consequently, applications are executed through WorkflowSteps.

Subject-oriented development consists of several phases: setup, refinement, validation, implementation assignments, and run-time:

- Setup – the configurator / administrator of the IoT-based CPS generates a digital behavior model (digital twin). Each concerned service and CPS component including IoT devices is represented by a behavior model (Subject Behavior Diagram).
- Refine representing behavior for each system node.
- Validate system behavior along messaging of individual subjects.

- Assign implementation details – after validation, organizational, and technological details need to be assigned to each subject to enable execution of a model.
- Run-time – during runtime (based on executable behavior models), additional IoT devices and services can send a registration message to become part of the CPS and get connected to other components or subjects.

This life cycle allows not only for design-integrated engineering, but also creating a digital representation of a CPS independently of the counterparts in the physical world – the subject-oriented specification contains run-time complete information on an implementation-independent layer of abstraction. In the assignment phase of development (i) the type of actor or system is defined that executes behavior models organization assignment), (ii) the technology is assigned to non-human CPS components, thus allocating physical components to digital behavior models. For instance, a traffic management system may be partially controlled by (i) humans, such as operation managers of smart cities, (ii) software, e.g., algorithms for optimizing the flow of traffic, and hardware, such as sensors recognizing cars for switching the traffic lights.

4.2 DT Thread Modeling

Based on the various DT concepts we first detail the DT thread understanding for the proposed approach before exemplifying it.

Figure 1 shows the DT thread concept when applying the behavior-centered subject-oriented modeling and execution technique. The digital model is accompanied by an execution unit that is connected to the cyber-physical components. Since models can be changed throughout runtime, design and engineering are integrated to control and adapt CPS operation. The CPS provides interactive access based on the models and delivers data either from users, sensors, or production systems. It also executes operation on the physical CPS part, thus delivers data and controls the cyber-physical components. Since subject-oriented modeling and execution has its origin in Business Process Management (BPM) (see [11], [35]) and currently there are some considerations to apply it to Industry 4.0 scenarios [38], the models also support the articulation of requirements for accomplishing tasks in socio-technical systems [39]. Design-integrated engineering means the refinement of models to the point of execution, as recently demonstrated in various contexts up to the Internet of Behaviors [34], [40].

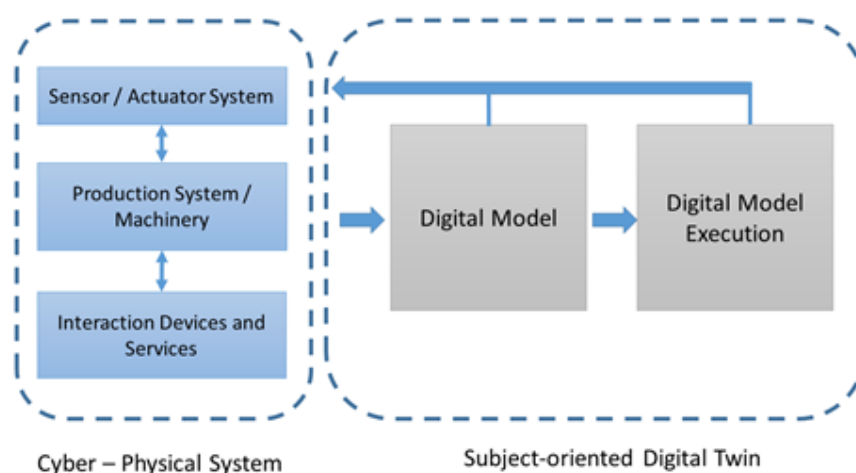


Figure 1. DT threads of CPS when applying subject-oriented modeling and execution

In a subject-oriented thread specification events are also represented by messages, since the only difference to sending/receiving regular messages lies in the type of transmitted information. Hence, a message can carry information when collecting data from sensors, conveying decisions, and exchanging business objects. Consequently, any CPS logic is considered in terms of

- structure, i.e. a set of subjects,
- flow of control, i.e. the exchange of messages, and the
- exchange of data, as they are part of the transmitted messages between subjects.

The realization aspects of the various subjects and the way they communicate with each other are considered in a step succeeding the specification of the business or production logic. For structured processing, messages are deposited at the receiver side in a so-called input pool. If a message is processed by the receiver it will be removed from its input pool. The protocol of communication is supported by the following logic of input pools. They have attributes, such as the size of an input pool which defines the number of messages that can be stored in the input pool in general. Another attribute is the number of messages of a specific type or from a certain sender that are allowed in a specific input pool.

Additionally, it can be defined whether messages are discarded upon receipt without further processing, or a sender is blocked if a message cannot be stored in an input pool when violating an attribute value. In an input pool there can be several messages of the same type from the same sender. They are distinguished by the sequence number indicating their arrival time. In case there is more than one message of a certain type in the input pool, the receiver will get the oldest one for processing. In this way, the specification of input pools allows covering requirements of both, the business operation at hand, and the technologies used for the implementation of a distributed system.

4.3 An Exemplary Use Case

We use a traffic management system as show case, based on several smart city studies performed in the field. Figure 2 as a demonstrator of the approach consists of nine participants represented by subjects which communicate with each other. The communication paths are shown by arrows. The collector subject counts cars reported by the four detector subjects and passes on the numbers to the traffic management subject, which exchanges data with the environment management. It also communicates with the intersection control 1, which in turn controls traffic light 1.

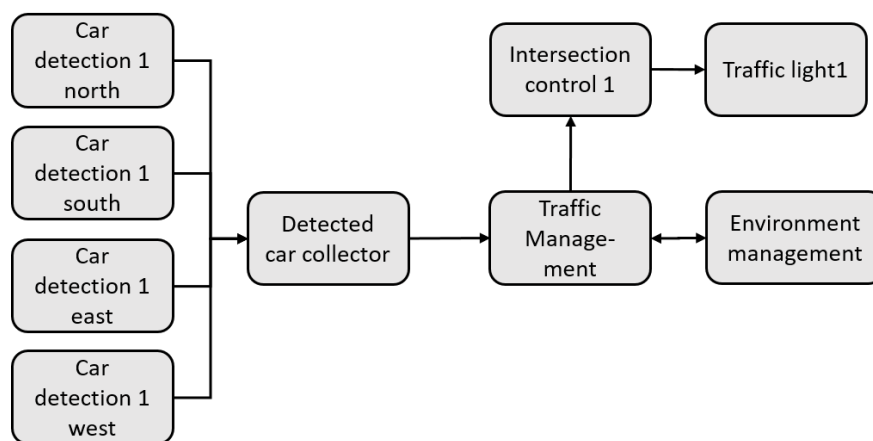


Figure 2. Communication Structure of a Simple Traffic Management System

The logical behavior of a subject is described by the allowed sequences in which messages are sent or received, and the actions executed on local objects/data. Therefore Figure 2 is enriched with the messages exchanged between the involved parties (Subjects). The diagram which shows the involved subjects and the messages they exchange (Figure 3) is called Subject Interaction Diagram (SID).

In a succeeding refinement step the behavior of each subject is defined. The Subject Behavior Diagram (SBD) specifies the allowed sequences in which messages are sent or received and data

are changed by internal operations of the subject. Figure 4 shows the behavior of the subject “Car detection 1 north” as an example. After receiving the message “switch on” the sensor starts detecting cars, which is represented by the internal action “detect cars”. If a car is detected, a message “car detected” is sent to the subject “Detected car collector”.

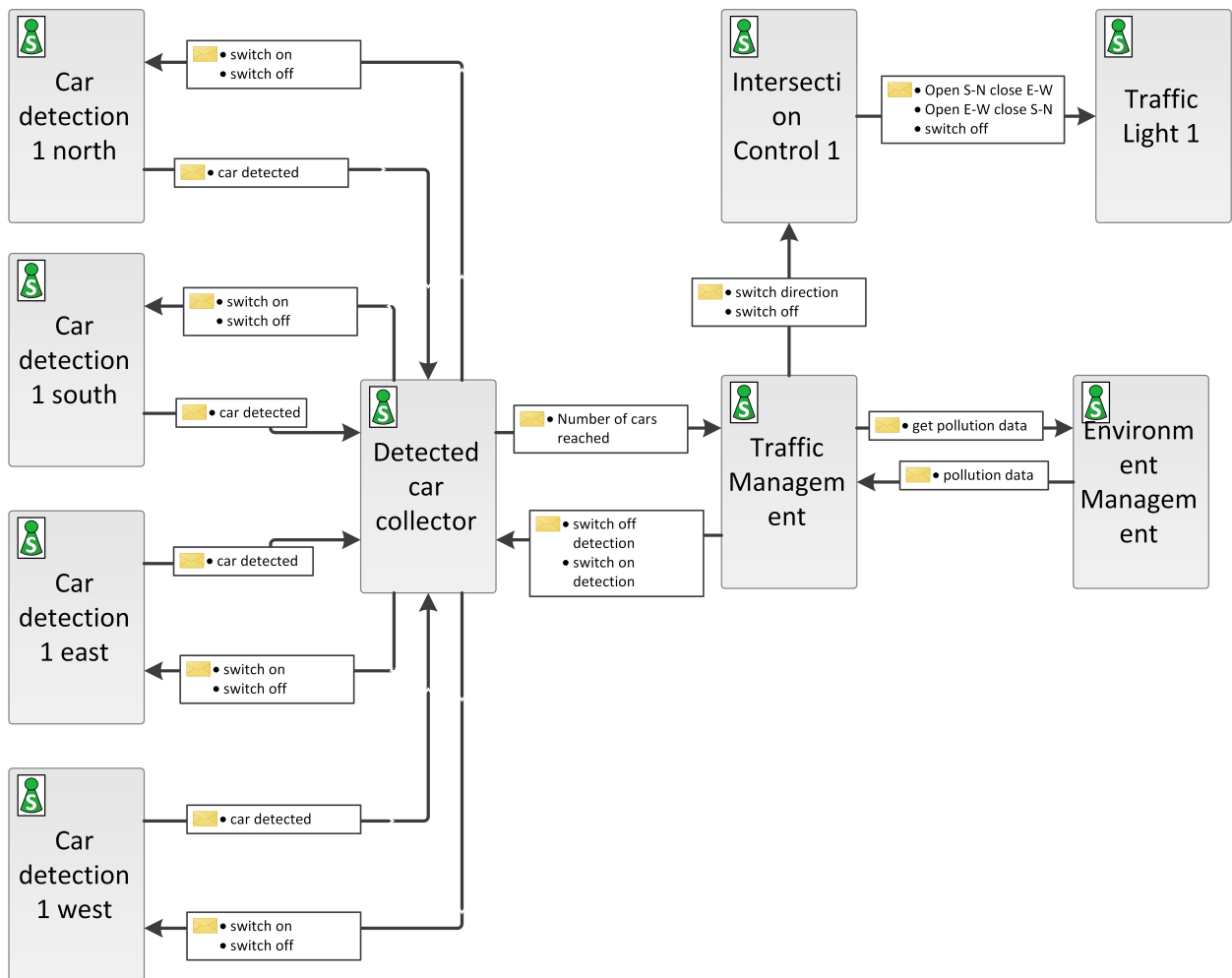


Figure 3. Subject Interaction Diagram of the Traffic Management System

The subject “Car detection1 north” can receive a message “switch off” anytime. Therefore it must be ready for receiving that message anytime. This is modeled by a so-called guard. Guards take care of high priority messages. When such a high priority message is deposited in the input pool, the current behavior sequence is interrupted, and the concerned subject switches its behavior to the guard. In our case the message “switch off” is accepted and the sensor switches to the state “end”. Guards are a mean to define the reactions to unpredicted and undesirable events.

From the perspective of the receiving subject, the message “car detected” is stored in the input pool of the subject “Detected car collector”. The subject “Detected car collector” picks up the messages from the car detectors, counts the cars and, informs the subject “Traffic Management” if a threshold is reached. The latter then controls the Intersection control 1, which in turn switches the subject “Traffic light 1” by sending the corresponding messages. Figure 5 shows the part of the behavior in which the subject “Detected car collector” receives the messages “car detected” from the various car detector subjects.

Due to the direct communication between subjects, each message must be received explicitly by a subject to be processed, and thus, it needs to be considered in the behavior description explicitly. Such a requirement can cause some inconveniences, e.g., once additional car detectors are added to the traffic management system. We assume another intersection that has to be managed,

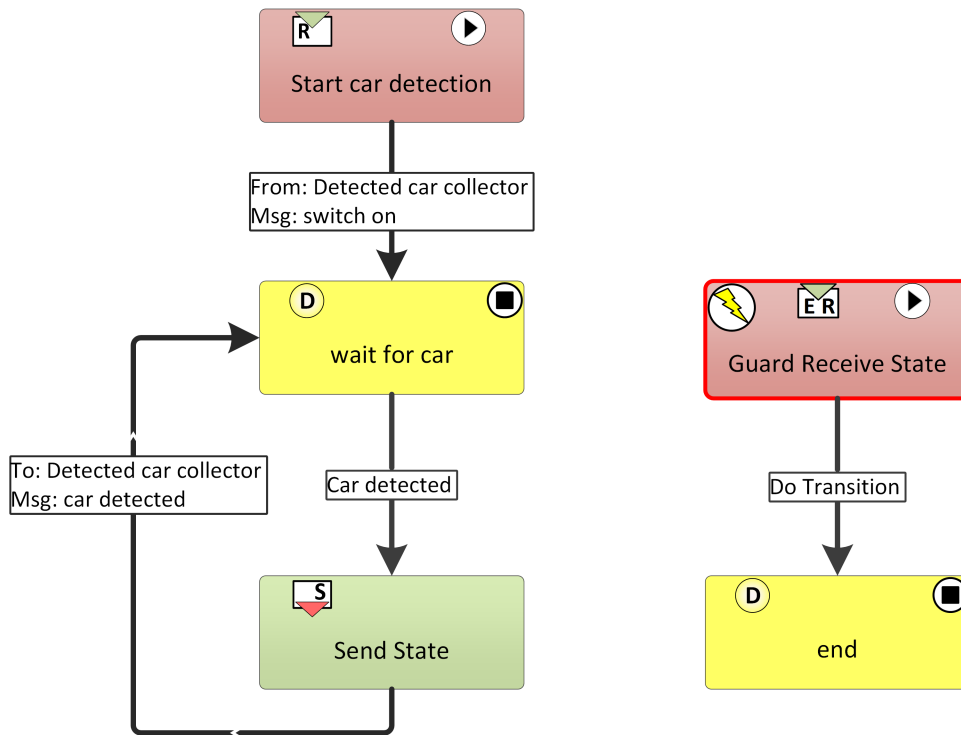


Figure 4. Behavior of Subject "Car detection 1 north"

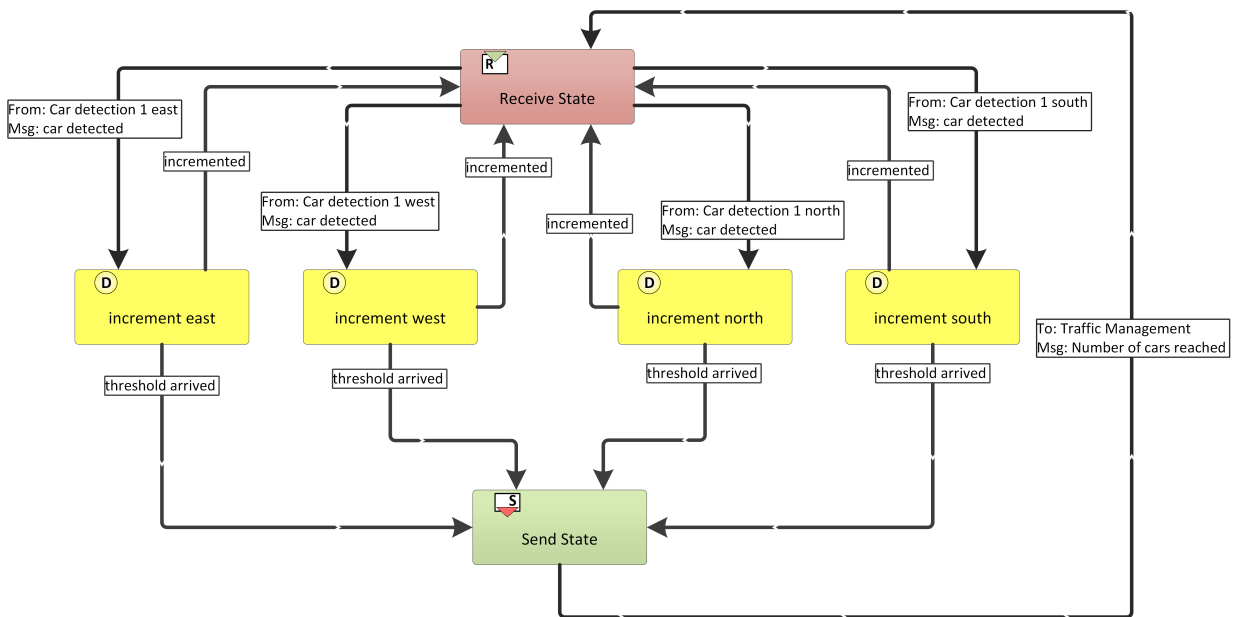


Figure 5. Part of the Behavior of Subject "Detected car collector"

however, provided with another four car detectors. Figure 6 shows the structure of the extended traffic management system. The example shows how an IoT system can be described by a process specification in a technology-independent way: The business/ program logic of a distributed system is detailed in terms of its component structure and its functional control flow and data exchange.

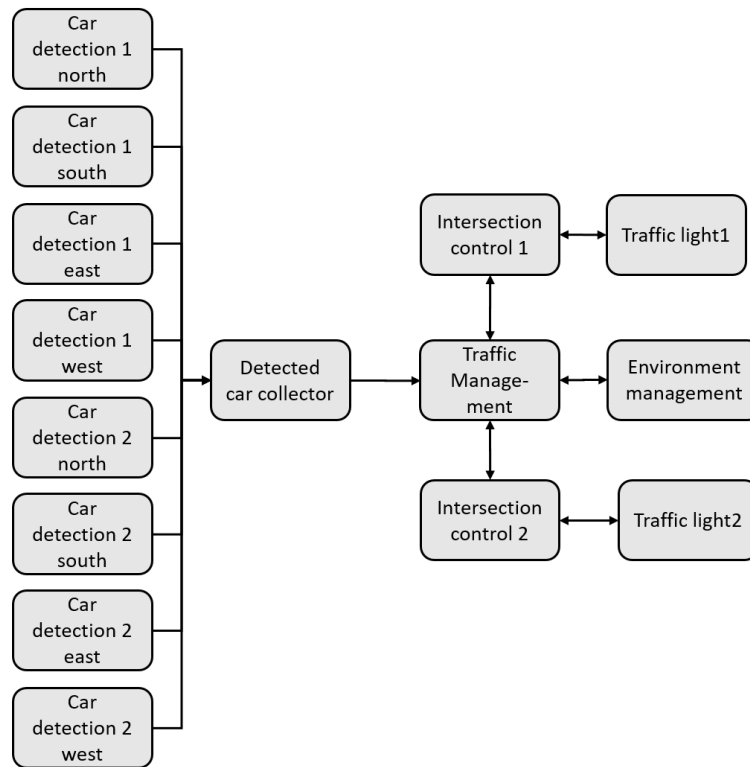


Figure 6. Extended Traffic Management System

4.4 Meeting Emergent Requirements

In this sub section we analyze several mechanisms that are relevant from a conceptual and application perspective, thus affecting design and engineering, as well as their integration for CPS development and evolving behavior utilizing DT threads:

- integration depth and breadth,
- connectivity capabilities,
- intelligent system behavior,
- validation capabilities,
- implementation variability from an organizational and technological perspective.

DT integration capabilities have been addressed by Yitmen et al. [41] with respect to business processes lately, indicating that business operation at a higher level of abstraction than IoT component interaction needs to be considered for design. From previous application studies (see [42]) we experienced both, the vertical, and the horizontal integration of processes need to be considered. For instance, if we consider the environment management to be integrated with traffic management, a vertical and horizontal integration makes sense. The vertical integration is required in case the behavior of the traffic management needs to be influenced by the environment management, e.g., in case of concentrated pollution in the air. The horizontal integration could be required on any level of detail when information between CPS nodes need to be exchanged, e.g., crossing sensors from traffic management delivering the concentration level to a pollution accumulator and calculator as part of environment management. With respect to connectivity,

subject orientation enables all system components (subjects) to be mutually connected. As demonstrated in [11], a system could be even designed by deconstructing the communication links between subjects or network nodes. However, designers have to bear in mind that each message can have different flavors:

- it carries on some data, e.g., sensor data from car detection, that need to be processed by the receiving subject, or business objects, such as a customer order;
- it informs as part of a decision making process, e.g., informing on the criticality of a measurement for additionally regulating traffic due to the level of air pollution.

In addition, subjects could be dedicated to specific tasks, such as control of air pollution, or taking care about privacy. In these cases the topology of a CPS as a set of networked nodes representing physical or digital components, or a combination of both, needs to be adapted. The System-of-Systems (SoS) perspective helps to implement dedicated features, such as pro-active environment management, represented as (interactive) access point for that task with essential business logic (see task list of emission gateway in Figure 7). This access point ‘Emission Gateway’ propagates dedicated messages either for pulling or pushing information through networks. It constitutes a temporary hierarchy and allows to enrich the operational intelligence (cf. [43]).

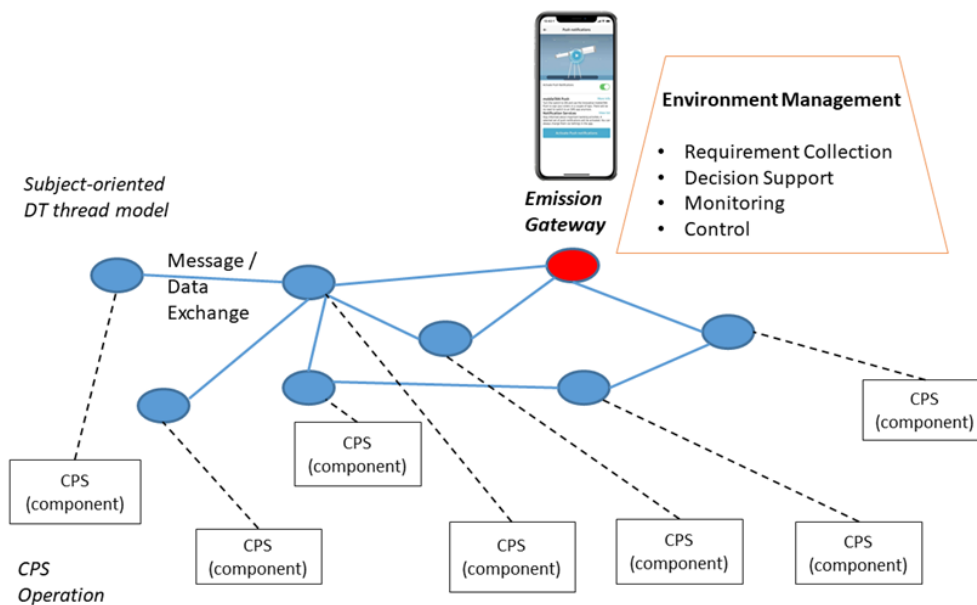


Figure 7. System-of-System constellation of integrated environment and traffic management

Emergent requirements and intelligent behavior can also occur in regular operations, and need to be captured by the DT model accordingly. For instance in order to take into account additional car detectors the DT modeler has to change the behavior of the subject “Detected car collector”. In order to couple the additional car detectors to the system, the communication with these detectors and the “Intersection control 2” subject have to be added. Such a minor extension of the overall setting leads to the necessity to reconsider all communication connections, in order to meet the requirements of controlling an intersection.

Hence, changing direct communication channels can be time-consuming and difficult, if not erroneous. In order to overcome that deficiency, we will extend the basic subject-oriented modeling concept with so-called shared input pools.

Shared input pools have the same structure like subject-specific ones, and thus, the same properties like the standard input pool. The only difference is that different subjects can deposit or remove messages from a shared input pool. Subjects that want to send a message via a shared input pool do not use a subject name as addressee of a message, but the name of a shared input pool instead.

In a distributed system several shared input pools for different purposes can be used. Figure 8 shows the slightly changed structure of the traffic management system when operating it with a shared input pool. The subject “Car detection” represents the shared input pool.

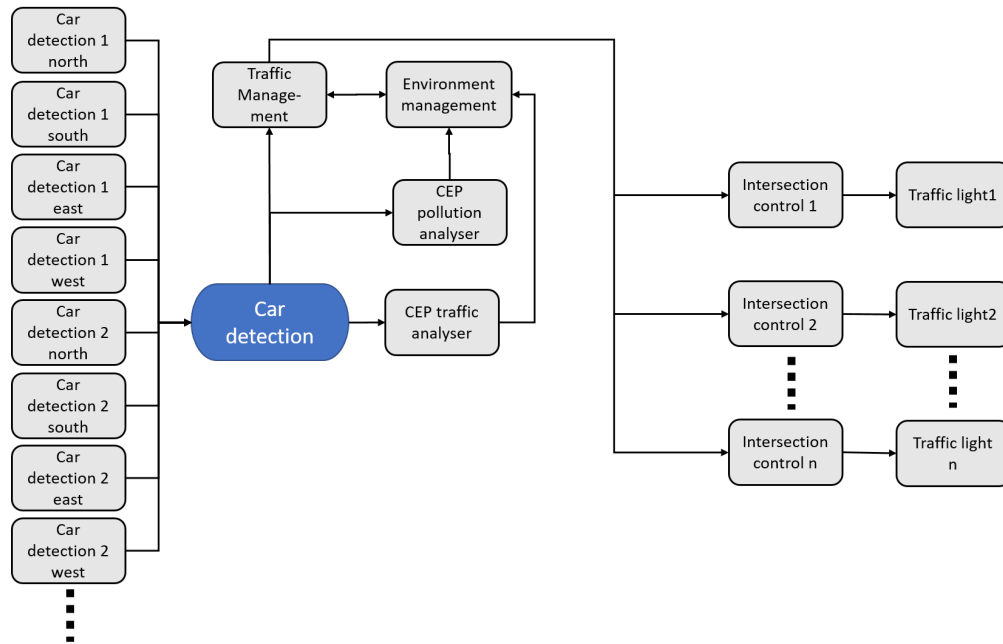


Figure 8. Traffic Management System with Shared Input Pools and Complex Event Processing

Shared input pools make a distributed system more flexible when additional participants or nodes are added. For instance, a third intersection control could be added to the traffic management system without much effort. In this case, only the additional detectors and the components for controlling the intersection have to be complemented and linked to the shared input pool. The extension would have no impact on the behavior of the other subjects and their behavior in that system.

There is one additional attribute for shared input pools: It defines whether a message will be removed from the input pool once a message has been picked up by a receiving subject. This mechanism is required, since several subjects may need to process a particular message. In addition, it allows keeping historical information in the input pool, in particular for analyzing the content of an input pool independently of the behavior of interacting subjects.

The messages of a shared input pool can be analyzed with respect to certain patterns of its messages. In order to perform such an analysis, Complex Event Processing (CEP) concepts can be applied. Complex Event Processing can be encapsulated in a subject. A subject of this kind scans the messages of a shared input pool and checks whether patterns of interest can be found. Once such a pattern is identified, a message including the discovered pattern can be sent to other participants, and initiate further activities. Figure 8 shows the traffic management example enriched with subjects processing complex events in order to reduce air pollution. Air pollution through street-bound traffic has been identified as one of the major environmental hazards for many regions and cities. A first step in handling traffic issues in the context of air pollution is the identification of source of emission, such as the growing rate of vehicle numbers and limits of average flow of vehicles on specific roads.

Based on these data, specific actions can be taken either on the macro- or micro-level, taking into account emissions and fuel consumption of the vehicles when considering the density, flow, and average velocity of traffic (cf. [44]). So-called moving-horizon approaches have led to implement variable speed limits and on-ramp metering rates to optimize fuel consumption and throughput times. Such systems require an adaptive twin and control inputs that are exchanged between controllers and cyber-physical regulation devices, such as traffic lights. Accordingly we can assume

that once the passing rate of vehicles at a traffic light falls below a certain threshold, e.g., due to overflow in one direction, that other components, such as previous traffic lights or ramp controllers are activated to change the metering rate for the concerned lane(s), or the speed limit is reduced to reduce the number of waiting vehicles. In our example, the CEP traffic and CEP pollution analyzer can identify respective threshold patterns and inform environment management which in turn provokes traffic management to change settings of traffic lights through intersection controls.

The situations captured include predictable behavior, with respect to both, desirable and undesirable behavior. For instance, in case the air pollution is low, traffic does not need to be restricted, in case the pollution level reaches a certain threshold, restrictions can be applied, either straightforwardly encoded in a subject behavior specification, or supported by complex event processing subjects.

In case unpredicted events occur, the DT thread needs to be adapted to proceed in terms of unpredictable behavior. Not only when modifications need to be implemented according to unpredicted events, but anytime when operation needs to be simulated, the subject-oriented DT specifications need to be validated. Validation means checking whether a behavior specification is effective, i.e., whether it yields the expected output in the form of a product or service. The subject of validation is the model, and thus the intended representation. It lays ground to optimization. Utilizing subject-oriented runtime engines, such as UeberFlow (based on the Akka framework <http://akka.io/>) [36], the Metasonic Suite (<https://www.metasonic.de/en/>) or the MS Visio based SiSi tool suite [37] the DT thread models can be executed without further model transformation (given that they are syntactically correct).

Keeping humans in the loop, stakeholders can specify and implement pro-actively, as well as adapt CPS behavior reactively by replacing behavior specifications within the limits of behavior diagrams during runtime. The screen shot in Figure 9 shows the use of the Sisi execution engine in order to first verify and then validate the behavior logic of the traffic management system.

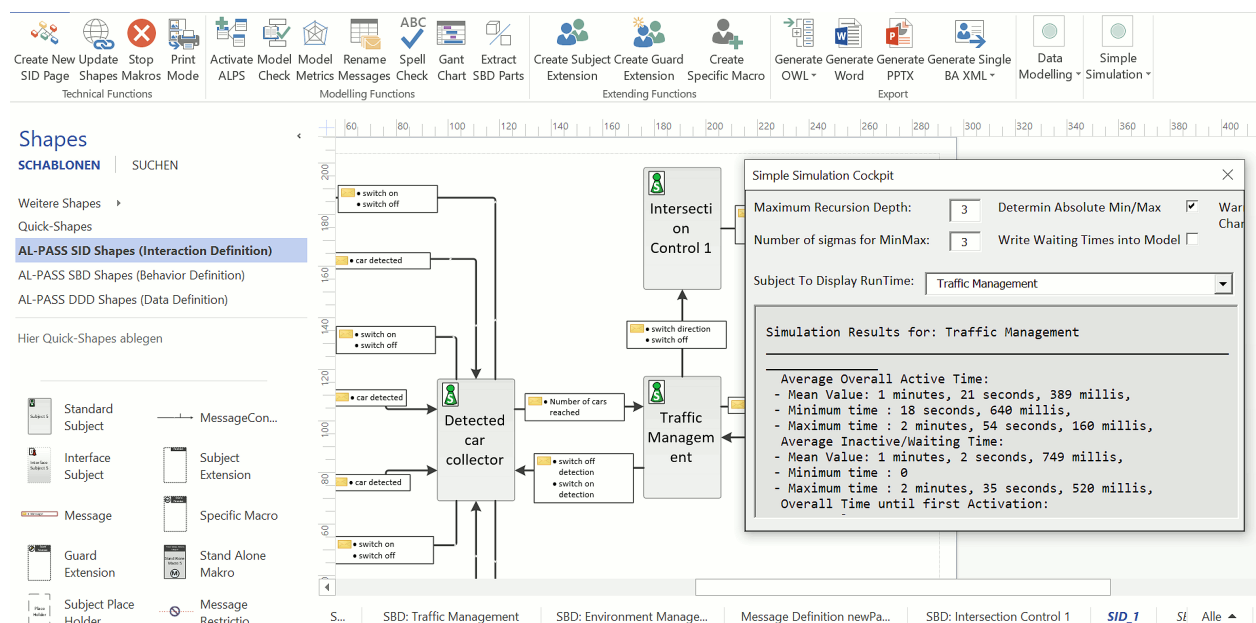


Figure 9. Screenshot of the SiSi-Execution Interface for Traffic Management (for one control cycle)

In this way, implementation variability can be achieved on the modeling level through design activities. The most important is the capability in S-BPM to separate models from organizational and technological implementation [11]:

- Organization-specific implementation. Once behavior specifications have been validated, it needs to be decided, how they become part of an existing or novel organizational environment.

The allocation of tasks to humans, technology, or hybrid systems has implications on the physical part of a CPS.

- Technological implementation. Once it has been decided, how validated behavior specifications become part of an organizational environment, either in terms of human role carriers, technological components, or hybrid systems, each component with some technology relation has to be assigned to a concrete machinery or technical system. Due to its nature, an IT-based workflow including the integration of a suitable user interface, business logic, and the required IT systems, is realized when subject-oriented modeling is applied.

Once implemented, behavior specifications, and thus processes, become productive (go live) in an organization. They are executed within the work structure of the organization and its IT environment in daily operations. In the course of monitoring process execution, data can be collected and recorded. They are calculated to provide accurate actual values to be compared with previously defined performance targets. The results are processed through reporting according to the need of target groups and made available to the intended recipients.

The evaluation of the results, when comparing actual performance data to plan data, may lead back to the analysis of causes in case of undesirable deviations, and depending on the nature of the perceived need for action, to the iteration of DT thread modeling. In this way, seamless and design-based CPS roundtrip engineering is implemented.

Each integrated design and engineering cycle may change the embodiment of digitally modeled CPS components into the physical reality, as indicated in Figure 10. Consequently, the CPS behavior can either be controlled by environmental triggers in the physical environment, or by digital means, involving human intervention at some point.

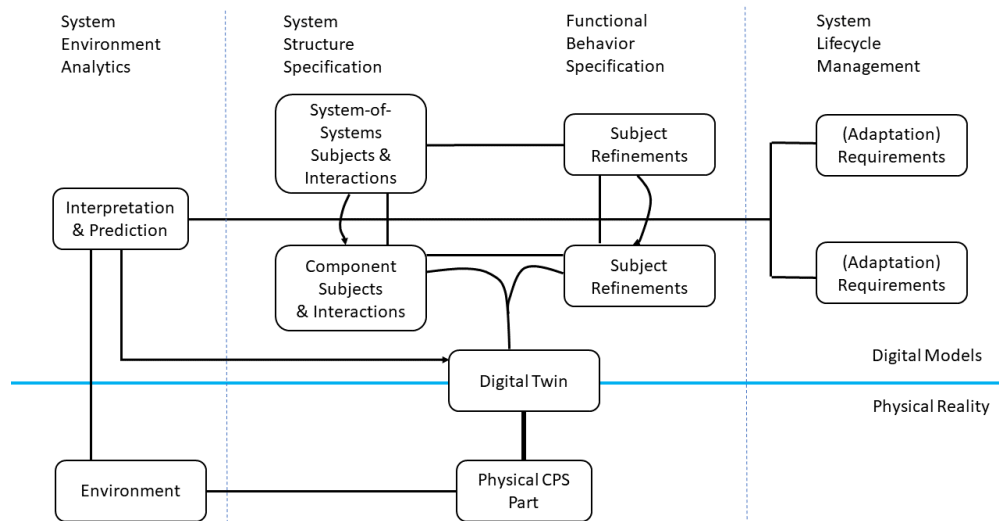


Figure 10. DT Embodiment in Physical Environment

4.5 Discussion of Findings

In this section we re-capture the requirements related to the various types of system behavior (Figure 2) and thus, dynamic DT-based CPS development, and summarize how behavior-centered modeling and execution can meet them utilizing subject-oriented specification and development steps.

The example use case could reveal essential properties of the subject-oriented approach to Design-integrated Engineering:

- Subject-oriented DTs enable integration of behavior in all 'directions' due to the network structure (represented by the SID on an abstract layer).
- The connectivity is standardized through message exchanging, and thereby, allows coupling of heterogeneous components.
- Complex operations can be decomposed in executable entities (subjects) that can be validated and simulated as part of domain-specific or organizational intelligence.
- Model validation facilitates checking semantic correctness of system components and their interplay.
- Implementation variability supported by subject-oriented development enables different organizational and technological variants of behavior specifications at runtime, and thus standardization of CPS behavior independent of implementation.

Table 2. Capturing Behavior Types for CPS Development through Design-Integrated Engineering

Behavior Categories	Predictable	Unpredictable
Desirable	The desired behavior of a CPS can be specified in terms of active components and intentional interaction design. The organizational and technology implementation allows to realize a system according to the physical and digital facilities and user interventions. The respective SID captures all relevant components to operate a CPS. For each subject of the SID a SBD provides behavior refinement in terms of functions and communication activities according to the SID interface specifications and the task to be accomplished.	The model contains 'pleasant surprises' by needs to adapt in terms of better user satisfaction and performance quality. Typical examples are messages that are not required for successful task accomplishment. Inputs for adaptation are generated when either validating the DT or operating it (coupled to the physical environment). 'Pleasant surprises' allow optimizing the CPS behavior. The DT facilitates optimization by enabling runtime validation and model execution when changing the SID and/or SBDs.
Undesirable	In DT-based CPS development some of these problems are known during modeling and thus, at design time, and can be captured by SBD message guards. In those cases the designer knows about these behavior exceptions and specifies variants of component behavior to reduce the likelihood of occurrence or eliminates it. P[redictable] U[ndesirable] problems can also occur in the course of organizational or technological implementation - management issues. They may lead to SID modification when the scope of behavior encapsulations needs to be changed.	Serious or catastrophic problems require model adaptation, validation, and run-time execution. They occur during runtime with the DT supporting ad-hoc adaptation of the SID and SBDs and their validation. Due to the choreographic arrangement local behavior modifications can help avoiding to change the overall CPS architecture, i.e., the SID. Moreover, decision support through Complex Event Processing specification can facilitate to judge whether and/or which changes should occur, referring to the overall structure (SID) or component behavior (SBDs).

5 Conclusion

DTs bring about the promise to help improve business operation and facilitate recognizing innovation opportunities in service and production industry through digital replicas of evolving support systems. Some challenges exist, for instance, delivering a high degree flexibility when creating and running a CPS ecosystem. They are due to the heterogeneity of networked components and the variability of behavior.

The proposed integrated subject-oriented modeling, validation, and execution support is based on coupling design and engineering in the course of DT development. Business operation and technology can be addressed from the perspective of specific stakeholders and concerned systems due to the abstraction as behavior entity or subject.

We have demonstrated several DT twin dimensions and concepts including an exemplary case study from traffic management. We could capture integration in-depth to operational system

functions and in-breadth when extending the scope of a CPS. Both integration dimensions are required, as fundamental operation as well as strategically relevant behavior needs to be captured in dynamic CPS settings. For DT behavior, entities connectivity is an essential enabler, as it facilitates cross-leveling, in particular when using a message-based interaction approach. The network nature enables not only coupling of behaviors to achieve a certain level of system intelligence, but also adapting system behavior, whenever novel requirements emerge.

The validation capabilities of behavior-oriented DT threads enable pre-checks and run-time execution before a CPS is instantiated. The introduced approach allows flexibility with respect to assigning human or technological actors to behavior entities. In this step, the organizational and technological perspective are adjusted.

Further case studies will have to be performed to explore the practicability of the approach, in particular in complex cross-domain challenges, such as traffic and pollution management, which could only been sketched in this work.

References

- [1] S. J. Berman, “Digital transformation: opportunities to create new business models,” *Strategy & Leadership*, vol. 40, no. 2, pp. 16–24, 2012. [Online]. Available: <https://doi.org/10.1108/10878571211209314>
- [2] S. Waschull, J. Bokhorst, E. Molleman, and J. Wortmann, “Work design in future industrial production: Transforming towards cyber-physical systems,” *Computers Industrial Engineering*, vol. 139, p. 105679, 2020. [Online]. Available: <https://doi.org/10.1016/j.cie.2019.01.053>
- [3] D. Karagiannis, R. A. Buchmann, X. Boucher, S. Cavalieri, A. Florea, D. Kiritsis, and M. Lee, “Omilab: A smart innovation environment for digital engineers,” in *Boosting Collaborative Networks 4.0*, L. M. Camarinha-Matos, H. Afsarmanesh, and A. Ortiz, Eds. Springer International Publishing, pp. 273–282, 2020. [Online]. Available: https://doi.org/10.1007/978-3-030-62412-5_23
- [4] M. Mühlburger, B. Krumay, and C. Stary, “Enterprise engineering for business opportunity recognition – a design science approach,” in *Proceedings 6th International workshop on managed complexity*. CEUR Proceedings, 2021.
- [5] A. Rasheed, O. San, and T. Kvamsdal, “Digital twin: Values, challenges and enablers from a modeling perspective,” *IEEE Access*, vol. 8, pp. 21 980–22 012, 2020. [Online]. Available: <https://doi.org/10.1109/ACCESS.2020.2970143>
- [6] M. Soltanifar, M. Hughes, and L. Göcke, *Digital Entrepreneurship: Impact on Business and Society*. Routledge/Springer Nature, p. 327, 2021. [Online]. Available: <https://doi.org/10.1007/978-3-030-53914-6>
- [7] G. Westerman, D. Bonnet, and A. McAfee, “The nine elements of digital transformation,” *MIT Sloan Management Review*, vol. 55, no. 3, pp. 1–6, 2014.
- [8] A. El Saddik, M. S. Hossain, and B. Kantarci, *Connected health in smart cities*, Springer Nature. 2020. [Online]. Available: <https://doi.org/10.1007/978-3-030-27844-1>
- [9] M. J. Kaur, V. P. Mishra, and P. Maheshwari, *The Convergence of Digital Twin, IoT, and Machine Learning: Transforming Data into Action*. Springer International Publishing, pp. 3–17, 2020. [Online]. Available: https://doi.org/10.1007/978-3-030-18732-3_1
- [10] D. Ivanov and A. Dolgui, “New disruption risk management perspectives in supply chains: digital twins, the ripple effect, and resilience,” *IFAC-PapersOnLine*, vol. 52, no. 13, pp. 337–342, 2019. [Online]. Available: <https://doi.org/10.1016/j.ifacol.2019.11.138>
- [11] A. Fleischmann, W. Schmidt, C. Stary, S. Obermeier, and E. Börger, *Subject-oriented business process management*, 2012. [Online]. Available: <https://doi.org/10.1007/978-3-642-32392-8>

- [12] T. Margaria, H. A. A. Chaudhary, I. Guevara, S. Ryan, and A. Schieweck, “The interoperability challenge: Building a model-driven digital thread platform for cps,” in *Leveraging Applications of Formal Methods, Verification and Validation*, T. Margaria and B. Steffen, Eds. Springer International Publishing, pp. 393–413, 2021. [Online]. Available: https://doi.org/10.1007/978-3-030-89159-6_25
- [13] J. Fryszak, *S-BPM Diagrams as Decision Aids in a Decision Based Framework for CPS Development*, ser. Communications in Computer and Information Science. Springer, vol. 1278, pp 23–32, 2020. [Online]. Available: https://doi.org/10.1007/978-3-030-64351-5_2
- [14] S. Boschert and R. Rosen, *Digital Twin—The Simulation Aspect*. Springer International Publishing, pp. 59–74, 2016. [Online]. Available: https://doi.org/10.1007/978-3-319-32156-1_5
- [15] F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee, “Digital twin in industry: State-of-the-art,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2405–2415, 2019. [Online]. Available: <https://doi.org/10.1109/TII.2018.2873186>
- [16] Y. Liu, L. Zhang, Y. Yang, L. Zhou, L. Ren, F. Wang, R. Liu, Z. Pang, and M. J. Deen, “A novel cloud-based framework for the elderly healthcare services using digital twin,” *IEEE Access*, vol. 7, pp. 49 088–49 101, 2019. [Online]. Available: <https://doi.org/10.1109/ACCESS.2019.2909828>
- [17] J. Bao, D. Guo, J. Li, and J. Zhang, “The modelling and operations for the digital twin in the context of manufacturing,” *Enterprise Information Systems*, vol. 13, no. 4, pp. 534–556, 2019. [Online]. Available: <https://doi.org/10.1080/17517575.2018.1526324>
- [18] R. Minerva, G. M. Lee, and N. Crespi, “Digital twin in the iot context: A survey on technical features, scenarios, and architectural models,” *Proceedings of the IEEE*, vol. 108, no. 10, pp. 1785–1824, 2020. [Online]. Available: <https://doi.org/10.1109/JPROC.2020.2998530>
- [19] N. Bagaria, F. Laamarti, H. F. Badawi, A. Albraikan, R. A. Martinez Velazquez, and A. El Saddik, *Health 4.0: Digital Twins for Health and Well-Being*. Springer International Publishing, pp. 143–152, 2020. [Online]. Available: https://doi.org/10.1007/978-3-030-27844-1_7
- [20] A. R. Hevner, S. T. March, J. Park, and S. Ram, “Design science in information systems research,” *MIS Quarterly*, vol. 28, no. 1, pp. 75–105, 2004. [Online]. Available: <https://doi.org/10.2307/25148625>
- [21] A. Hevner and S. Chatterjee, *Design Science Research in Information Systems*, pp. 9–22, 2010. [Online]. Available: https://doi.org/10.1007/978-1-4419-5653-8_2
- [22] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, “A design science research methodology for information systems research,” *Journal of Management Information Systems*, vol. 24, no. 3, pp. 45–77, 2007. [Online]. Available: <https://doi.org/10.2753/MIS0742-1222240302>
- [23] C. Cimino, E. Negri, and L. Fumagalli, “Review of digital twin applications in manufacturing,” *Computers in Industry*, vol. 113, Article 103130, 2019. [Online]. Available: <https://doi.org/10.1016/j.compind.2019.103130>
- [24] R. Wagner, B. Schleich, B. Haefner, A. Kuhnle, S. Wartzack, and G. Lanza, “Challenges and potentials of digital twins and industry 4.0 in product design and production for high performance products,” *Procedia CIRP*, vol. 84, pp. 88–93, 2019. [Online]. Available: <https://doi.org/10.1016/j.procir.2019.04.219>
- [25] M. Grieves and J. Vickers, *Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems*. Springer International Publishing, pp. 85–113, 2017. [Online]. Available: https://doi.org/10.1007/978-3-319-38756-7_4
- [26] K. Y. H. Lim, P. Zheng, C.-H. Chen, and L. Huang, “A digital twin-enhanced system for engineering product family design and optimization,” *Journal of Manufacturing Systems*, vol. 57, pp. 82–93, 2020. [Online]. Available: <https://doi.org/10.1016/j.jmsy.2020.08.011>

- [27] Q. Xu, S. Ali, and T. Yue, “Digital twin-based anomaly detection in cyber-physical systems,” in *2021 14th IEEE Conference on Software Testing, Verification and Validation (ICST)*, pp. 205–216, 2021. [Online]. Available: <https://doi.org/10.1109/ICST49551.2021.00031>
- [28] S. Ahleroff, X. Xu, R. Y. Zhong, and Y. Lu, “Digital twin as a service (dtaas) in industry 4.0: An architecture reference model,” *Advanced Engineering Informatics*, vol. 47, Article 101225, 2021. [Online]. Available: <https://doi.org/10.1016/j.aei.2020.101225>
- [29] J. C. Kirchhof, J. Michael, B. Rumpe, S. Varga, and A. Wortmann, “Model-driven digital twin construction: Synthesizing the integration of cyber-physical systems with their information systems,” in *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems*, ser. MODELS ’20. Association for Computing Machinery, pp. 90–101, 2020. [Online]. Available: <https://doi.org/10.1145/3365438.3410941>
- [30] T. C. Zimmermann, C. Masuhr, and R. Stark, “Mbse-entwicklungsfähigkeit für digitale zwillinge,” *Zeitschrift für wirtschaftlichen Fabrikbetrieb*, vol. 115, no. s1, pp. 51–54, 2020. [Online]. Available: <https://doi.org/10.3139/104.112312>
- [31] K. Sandkuhl and J. Stirna, “Supporting early phases of digital twin development with enterprise modeling and capability management: Requirements from two industrial cases,” in *Enterprise, Business-Process and Information Systems Modeling*, S. Nurcan, I. Reinhartz-Berger, P. Soffer, and J. Zdravkovic, Eds., pp. 284–299, 2020.
- [32] F. Tao, F. Sui, A. Liu, Q. Qi, M. Zhang, B. Song, Z. Guo, S. C.-Y. Lu, and A. Y. C. Nee, “Digital twin-driven product design framework,” *International Journal of Production Research*, vol. 57, no. 12, pp. 3935–3953, 2019. [Online]. Available: <https://doi.org/10.1080/00207543.2018.1443229>
- [33] C. Wu, Y. Zhou, M. V. Pereira Pessôa, Q. Peng, and R. Tan, “Conceptual digital twin modeling based on an integrated five-dimensional framework and triz function model,” *Journal of Manufacturing Systems*, vol. 58, pp. 79–93, 2021. [Online]. Available: <https://doi.org/10.1016/j.jmsy.2020.07.006>
- [34] A. Fleischmann, W. Schmidt, and C. Stary, *S-BPM in the Wild: Practical value creation*, 2015. [Online]. Available: <https://doi.org/10.1007/978-3-319-17542-3>
- [35] A. Fleischmann, S. Oppl, W. Schmidt, and C. Stary, *Contextual process digitalization: changing perspectives—design thinking—value-led design*, p. 275, 2020. [Online]. Available: <https://doi.org/10.1007/978-3-030-38300-8>
- [36] F. Krenn and C. Stary, “Exploring the potential of dynamic perspective taking on business processes,” *Complex Systems Informatics and Modeling Quarterly*, vol. 8, no. 2, 2016. [Online]. Available: <https://doi.org/10.7250/csimq.2016-8.02>
- [37] M. Elstermann and J. Ovtcharova, “Sisi in the alps: A simple simulation and verification approach for pass,” in *Proceedings of the 10th International Conference on Subject-Oriented Business Process Management*, ser. S-BPM One ’18. Association for Computing Machinery, 2018. [Online]. Available: <https://doi.org/10.1145/3178248.3178262>
- [38] A. Fleischmann, A. Friedl, D. Großmann, and W. Schmidt, *Modeling and Implementing of Industrie 4.0 Scenarios*, ser. Communications in Computer and Information Science, A. Dahanayake, O. Pastor, and B. Thalheim, Eds. Springer, vol. 1401, 2021. [Online]. Available: https://doi.org/10.1007/978-3-030-72696-6_4
- [39] S. Oppl and C. Stary, *Designing Digital Work: Concepts and Methods for Human-centered Digitization*. Routledge/Springer Nature, 2019. [Online]. Available: <https://doi.org/10.1007/978-3-030-12259-1>
- [40] C. Stary, “The internet-of-behavior as organizational transformation space with choreographic intelligence,” in *Subject-Oriented Business Process Management. The Digital Workplace –*

Nucleus of Transformation, M. Freitag, A. Kinra, H. Kotzab, H.-J. Kreowski, and K.-D. Thoben, Eds. Springer International Publishing, pp. 113–132, 2020. [Online]. Available: https://doi.org/10.1007/978-3-030-64351-5_8

- [41] I. Yitmen, S. Alizadehsalehi, I. Akiner, and M. E. Akiner, “An adapted model of cognitive digital twins for building lifecycle management,” *Applied Sciences*, vol. 11, no. 9, 2021. [Online]. Available: <https://doi.org/10.3390/app11094276>
- [42] M. Neubauer and C. Sary, *S-BPM in the Production Industry: A Stakeholder Approach*. Springer International Publishing, p. 232, 2017. [Online]. Available: <https://doi.org/10.1007/978-3-319-48466-2>
- [43] C. Sary, “System-of-systems design thinking on behavior,” *Systems*, vol. 5, no. 1, 2017. [Online]. Available: <https://doi.org/10.3390/systems5010003>
- [44] N. Groot, B. De Schutter, and H. Hellendoorn, “Integrated model predictive traffic and emission control using a piecewise-affine approach,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 2, pp. 587–598, 2012. [Online]. Available: <https://doi.org/10.1109/TITS.2012.2227314>