Decision-Making About Federated Digital Twins – How to Distribute Information Storage and Computing

Leif Bonorden[1] Marc Frerichs[2] Matthias Riebisch[1] Stephanie von Riegen[3] Florian Hartke[4] Rainer Herzog[3] Lothar Hotz[3] Dennis Jürgensen[2] Markus Kiele-Dunsche[5] Seeko Schottler[4] Rafael Schroeder[6]

Abstract: Digital Twins are commonly used as virtual representations of physical objects in manufacturing industries. Information for Digital Twins may be collected from multiple sources and stored in a distributed manner, leading to a Federated Digital Twin. Since decisions about such a federation are crucial for the system and its architecture, they should be guided by reliable and well-evaluated methods. However, current research is focused on distributed data sources but is missing decisions about the distribution of the digital twin itself. We present an approach to partition Federated Digital Twins by classifying information types, computing resources, and concerns of data suppliers. Furthermore, we show how decisions are made based on the Decision Model and Notation standard and evaluate the approach using an industrial case study.

1 Introduction

The idea of a digital twin (DT) addresses the need to digitally reflect the complete life cycle of physical objects, from planning to construction, operation and discontinuation. Different kinds of data, from design drawings and current runtime data to configurations and order data, can be accommodated in corresponding submodules [DI16; Pl18] of the DT. These data are then easily accessible for engineering, for example, to achieve more efficient utilization or optimize parameterization. The ADAM project[7] in the field of factory automation is concerned with the automated adaptation of production plants using a DT. In the project, concerns have arisen about the visibility and sharing of critical data such as configuration, production and order data. In addition, production plants produce a tremendous amount of data, the processing of which is intended as part of the digital transformation but is not easily possible with the resources of plants with some legacy elements. A monolithic DT is not sufficient to resolve these concerns, thus, federation of the DT is necessary.

---

[1] Universität Hamburg, Department of Informatics, {leif.bonorden, matthias.riebisch}@uni-hamburg.de

[2] Universität Hamburg, Hamburg Research Center for Information Systems, {marc.frerichs, dennis.juergensen}@uni-hamburg.de

[3] HITeC e.V., {stephanie.von.riegen, rainer.herzog, lothar.hotz}@hitec-hamburg.de

[4] encoway GmbH, DOCK ONE, {florian.hartke, seeko.schottler}@encoway.de

[5] Lenze SE, markus.kiele-dunsche@lenze.com

[6] Friedrich Remmert GmbH, rsc@remmert.de

[7] `https://uhh.de/adam-projekt`

Digital twins are usually defined as virtual representations of physical objects that are continuously updated with changes in properties or behavior. In manufacturing industries, DTs are commonly used to monitor and optimize production processes, identify and mitigate unexpected events, and test and simulate conditions [BCF19; Gr14; Se21]. In particular, DTs tightly link virtual and physical counterparts by representing structures (e.g., subcomponents, physical properties), a current configuration (e.g., settings, firmware), and a current state (e.g., voltage, speed). In ADAM, a DT is available for demonstration purposes in a prototype of some components of the plant as well as for the whole plant. This DT contains, among other things, product information, dimensions, configuration information (e.g., settings, firmware), and even design data in some corresponding submodels.

However, this DT is built as a monolithic one, leading to three main negative concerns:

C1 data ownership, access rights, and intellectual property of all parties involved need to be preserved,

C2 computing power and storage capacity are limited in most units,

C3 the volume of communication data might exceed data transmission capabilities between units.

Therefore, a distribution into a federated DT (FDT) has to be performed. Even if this task is primarily concerned with the distribution of data, we are aware that this also leads to a distribution of data processing.

To generalize the decision-making process, this paper's contribution is a decision model (DM) for dividing a DT and distributing the FDT to computing units. The DM considers multiple goals and constraints, such as data ownership concerns, resource capacity constraints, and availability goals. The DM is formalized via the standard Decision Model and Notation (DMN). We organize our work along with the following research questions:

RQ1 Which criteria are relevant to split a DT?

RQ2 How can the DT parts be assigned to system parts?

RQ3 What is a general (formalized) decision process to distribute a DT?

The resulting DM has to fulfill the goals of clarity, understandability, mastering complexity, appropriateness, adaptability, and generalizability and transferability to other domains. The notation has to be flexible, understandable, and appropriate. Many of these goals demand an evaluation in a real-world context. We want to stress the point that there are two notions of the term *Model* in this paper: firstly, the DT represents a model of a system and its parts, and secondly, a DM is developed and discussed throughout this paper.

The remainder of this paper is structured as follows: After relating our approach to other works in Sect. 2, we introduce the case study in Sect. 3. In Sect. 4, we present the research methodology Design Science Research and apply it to develop a decision model using the case study as an illustrative example. Subsequently, we evaluate the result in Sect. 5, and summarize the paper in Sect. 6.

2 Related Work

Eramo et al. consider a *digital twin* as a “virtual representation of an actual system” [Er22]. Thus, DTs are models themselves or contain models among data and possibly other information, e.g., historic information and predicted behavior [Er22; WD20]. Furthermore, five significant properties of DTs are commonly considered [Se21]: A DT mirrors the actual system’s entire life cycle. A DT obtains and includes real-time data. A DT is synchronized with the existing system. A DT includes behavioral information about the existing system. A DT provides means for interaction.

The construction of a DT from multiple models and various data sources has been discussed in research, e.g. [KMM20; Sc17; Ta19], but usually these approaches aim to construct a single, monolithic DT. To the best of our knowledge, no attempts to split the DT into an FDT have been conducted.

Digital Shadows are a related concept for describing, abstracting, aggregating, and connecting data from Digital Twins [Be21], modeling a one-way data flow with the state of an existing physical object. However, since ADAM is supposed to influence the physical object automatically, this concept is not applicable here.

The integration of models from multiple engineering disciplines precedes the concept of DTs, e.g., with AutomationML, which is now commonly used in the implementation of DTs [Sc16; Se21].

To model our approach to FDTs, we use the *Decision Model and Notation* (DMN) standard [Ob21] because of its openness and wide acceptance. An alternative would be *The Decision Model* [HG09]. However, it is owned by a specific vendor and subject to restricted distribution. All expressions are limited to using simple values or defining their syntax, unlike FEEL of DMN. Another alternative is given by the *IBM Operational Decision Manager* [IB22], but it requires a license and bears the risk of discontinued support by the vendor.

3 Case Study

We develop and evaluate the decision model in the factory automation domain, focusing on the construction and operation of a production system where different physical components are used, together with a DT. A significant use case consists of adaptations of a production system due to changes in products or plans, which demand changes in production systems, each consisting of different machines, components, or configurations. These changes are triggered by changing requirements or sensor data, and they are driven by engineering recipes out of a solution library of a component provider or a plant manufacturer.

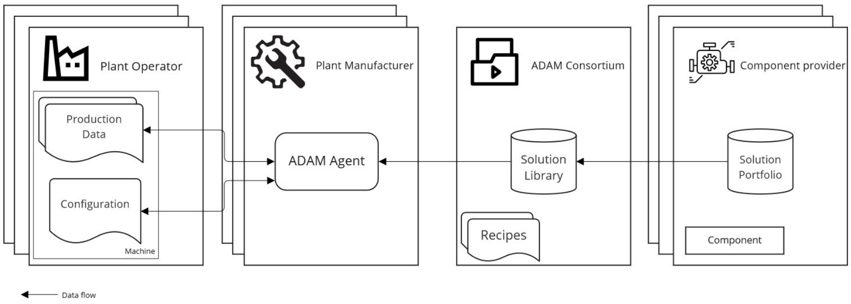


Fig. 1: ADAM stakeholders and structure

First, we introduce a typical use case, and then we give more details on stakeholders and structure, with Fig. 1 illustrating. As a use case for adaptation, we consider a production system in operation. Its configuration is set up for “sheet metal separation” consisting of components of different types like the transport units, e.g., conveyor belts, drive units, and sensors. After a new order has been accepted, the processed material changes to heavier metal sheets. This change demands higher force and load on some of the production system’s components, e.g., some drive units must provide higher torque. Therefore, an adaption of the production system is needed to increase the plant’s capabilities.

To perform this adaptation, the plant operator needs support from the plant manufacturer. However, the required higher torque demands a change of the drive provided by a component provider. This component provider gets the needed parameters and suggests a new drive solution consisting of an electrical drive with a parameter set. The suggestion by the component provider is developed using recipes. With a recipe, engineering expertise is made available for integrating the new solution into the production system. The ADAM agent (see Fig. 1) monitors sensor data. In a case of unsatisfied needs, and performs a lookup for suitable engineering solutions from plant manufacturers and component providers (solution libraries). For selecting appropriate recipes, configuration data of the production plant, the machines, and their components have to be assessed.

The plant and its machines produce a tremendous amount of sensor data, which leads to a high update frequency of data, which could cause limitations in transmission and processing resources in the different parts of the system, which refers to concerns C2 and C3 (see Sect. 1). These data could be filtered, not hampering their usage as a trigger for changes or adaptations. Moreover, the control computers’ storage capacity and computing power in currently existing plants are mostly too small for extensive computing, e.g., by a DT, which would increase these concerns.

Further data is needed to perform the adaptation task as an ADAM result. For parts of them, some of the stakeholders have concerns regarding privacy, which refers to concern C1 (see Sect. 1). Information about the current drive units is held within the configuration model of the production system in the DT. Data about torque and speed is covered by the DT as well. The configuration model might be stored in the production system, but in our case, the plant manufacturer is concerned about their release as intellectual property and critical for competition. Order information such as scope and execution shall not be released to other orderers for competition and privacy reasons. Plant operators do not want to disclose their capacity utilization because it might affect negotiations about future contracts and pricing. Therefore, historical data must be considered particularly critically. Additionally, access to the production plant, the machines, and the plant control station from the outside is not desired to avoid disruptions to the production process. Recipes for drive application and parameterization are considered as engineering expertise, and therefore, as business secrets of the drive component provider.

4 Approach

Tab. 1: Example separation between units

Unit and Location	**Data**	**Computing**	**Part of DT**
A machine	with process events, measured values for heat, torque, max rotation, speed of drives, error logs	preprocessing of data	production data, structure information, runtime data
B production system	history of process events, current order data	data aggregation	aggregated machine and current order data
C plant control station	history of order data	planning, monitoring, filtering, aggregation	scheduling of orders, capacity and utilization, error messages from production, maintenance requests, requests for change
D ADAM Agent	list of observed Trigger data sets, candidate solutions for alternative drive units	decision-making on adaptation with computing solutions	decision-making on adaptation, selection from a set of proposed adaption solutions
E drive (solution) provider	sample cases for application of drive units	filtering	set of proposed solutions for a variety of applications and configurations

The initial situation in the ADAM research project (see Sect. 3) is applied as a starting point to explain the decision-making and to elaborate the approach.

Initially, there is a prototype that visualizes and simulates components within parts of machines. A DT is used as the model driving visualization and simulation. Some components are even integrated as real hardware elements. The prototype is used for demonstration, e.g. in an exhibition. Some of the DT data are imported from engineering tools – e.g., for electrical drive configuration.

While the initial DT is a monolithic one, its transformation into a real plant is not an option because of the concerns introduced in Sect. 1. Therefore it has to be split and distributed, answering the three research questions (see Sect. 1).

For illustration purposes, Tab. 1 shows a separation into different operational units with respect to computing and aggregating data. The units correspond to the case study's architecture. The distinctions between the units that lead to this example separation are:
A – B: Memory limitation of the control computer.
B – C: Visibility of order data of different orderers to others.
C – D: Insufficient computing capacities for data analytics to compute adaptations.
D – E: Non-disclosure of business secrets of component providers, therefore filtering configuration data according to actual component installation at a plant.

4.1 Basic Methodology

Design Science Research (DSR) aims to develop new results (so-called artifacts) through a practice-oriented research approach and make them usable [He04]. The research presented follows the DSR methodology. Österle et al. [Ös11] suggest a framework that divides DSR into analysis, design, evaluation, and diffusion phases.

For the design of the proposed artifacts, the authors primarily used the design principles of induction [KGM12], and prototyping [NJ82]. In DSR, design principles (DPs) are "generic, high-level representations of the class of solutions addressing a class of problems" [Ko16], and they are derived from meta-requirements (MRs). The DSR methodology covers three essential steps of design principles induction: (1) deduce (meta-)requirements, (2) distill adequate DPs, and (3) formulate design decisions (DDs) as instantiations of DPs. Consequently, this leads to a one-to-many deduction from DPs to DDs [KGM12].

We use the Decision Model and Notation (DMN) [Ob21] standard to depict the decisions adequately. Published by the Object Management Group (OMG), this widely accepted and well-documented open standard creates the possibility of documenting decision rules. Furthermore, the adoption of this standard might increase applicability in other domains. Sect. 4.4 describes the adoption of DMN in more detail.

4.2 Eliciting Requirements for Decision-Making

We discuss the requirements according to the concerns **C1–C3** as introduced in Section 1.

C1 There are stakeholders with different roles such as plant operators, component providers, and plant manufacturers involved, each with their own concerns for their realm — especially about sharing relevant but critical data (process data, production data). The stakeholders are aware that they not only have to protect the data but also to release them partially to benefit from their processing. Therefore, this is a criterion for splitting the DT into partitions (RQ1). According to RQ2, the DT partitions have to be assigned to the realms of each stakeholder.

C2 There are computing devices with limited resources, namely network, storage capacity and computing power. This is especially urgent for legacy devices. The available resources of the components, machines, or system parts have to suffice the needs to process the amount of data. According to RQ1, the DT has to be split into partitions. These partitions are then assigned in a way that the needed resources does not exceed the available ones (RQ2).

C3 Network connections between parts differ according to speed, capacity and even robustness. For periodic updates there is a certain need for data transmission. The DT must not be split into partitions in a way that the needed transmission between them would exceed the available capacities (RQ1).

As the first step of design principles induction, meta-requirements have been collected based on the concerns mentioned before (C1–C3). Aiming at a DM for the distribution of a DT (RQ3), these meta-requirements need to be transferred into DPs, thus, representing another step of our DSR approach. The meta-requirements, together with the corresponding DPs, are given in Tab. 2.

Regarding meta-requirement MR1, derived from concern C1, it can be concluded that this requirement has a high priority because the respective owner only shares data if privacy is assured and/or if need and benefit are accepted. This is always a trade-off between risk and need for the data owner. To fulfill this requirement and keep the hurdle for sharing data low, sensitive data should remain locally with the owner. External access should only be possible via appropriate interfaces and following the granted access rights (DP1). Tab. 2 shows the DPs and the underlying MRs identified based on the concerns described above.

Meta-requirement MR2 addresses concern C2. We propose splitting the DT into subsystems when computing needs or storage capacity require it (DP2). To resolve conflicts of needs vs. resources, a proper assignment of tasks to the existing systems must also be achieved.

To fulfill meta-requirement MR3, which is based on concern C3, an appropriate computing and distribution setup is proposed: The update rate and data aggregation should be defined following the given resources (DP3). Furthermore, the amount of data requested should also be reduced according to the resources.

Tab. 2: Design principles and meta-requirements

	Design Principle	Meta-requirement
DP1	Make sure that the *sensitive data is located locally on the owner's system* — access only via *defined interfaces* and according to *granted permissions*.	*MR1* Data ownership, privacy, and intellectual property must be guaranteed at all times.
DP2	Use *subsystems* if the need for computing power or storage capacity requires it and *assign the tasks properly*.	*MR2* The computing power must be sufficient for the chosen purpose. The needs must be satisfied by the resources.
DP3	Use *appropriate computing and distribution setup*, *set update rates*, and *aggregate data* according to the given resources.	*MR3* Computing and distribution must be carried out even on (and for) systems with limited resources.

4.3 From Design Principles to Design Decisions

According to the DSR procedure, the next step is to convert the DPs into concrete DDs that form the basis for the FDT. Tab. 3 summarizes the DDs and their interrelation to the DPs from which they are derived.

Tab. 3: Design decisions derived from design principles

	Design Decision	DP(s)
DD1	Local data service accesses parts of DT	DP1
DD2	Interfaces with certified/trusted components	DP1
DD3	Data transmission in aggregated or anonymized form	DP1
DD4	DT parts access and accessed by distributed services/subsystems	DP1, DP2
DD5	Federated computing & storage	DP1, DP2, DP3
DD6	Proper assignment of tasks to subsystems	DP2, DP3
DD7	Caching & Queuing	DP3
DD8	Data aggregation & Sampling rate reduction	DP3

Based on DP1, we conclude that data services providing sensitive data should be designed to access local parts of the DT (DD1). Since the data exchange corresponds to an upstream-downstream relationship, it could—depending on the use case—be designed using the customer-supplier or conformist pattern [Ev04]. In addition, we propose the use of well-defined and purpose-driven interfaces that only grant specific permissions for data access to certified or trusted components (DD2). Thus, data ownership, privacy, and intellectual property protection are considered. Where applicable, the data transmission should occur in aggregated or anonymized form instead of raw data, thus, related to the principles of data economy and data adequacy (DD3). Machine operators and manufacturers have to perform a risk/benefit analysis which data should be shared.

The second meta-requirement, on which DP2 is based, addresses the different computing requirements and resources of the various system levels. DP2 proposes the use of subsystems and a proper assignment of tasks to the respective systems related to both RQ1 and RQ2. Thus, we conclude that the DT should consist of distributed services/subsystems (DD4) with an appropriate needs and resources balance (DD6). The requirement for federated computing and storage (DD5) is derived following DPs 1–3.

Caching and queuing (DD7) is proposed as a design decision related to DP3. These techniques, e.g., control load on the respective systems. Data aggregation and sampling rate reduction (DD8) are also means of adapting the amount of data to the resources of the respective systems and the actual data requirements. Message queues or producer-consumer patterns are proposed to accomplish distributed computing and storage, queuing, or aggregation since they can process, transform and distribute data streams.

Fig. 2 gives an exemplary overview of the application of design principles induction, starting with the three concerns up to derived design decisions.

	C1: Data Ownership & Data Privacy	**C2: Computing Power & Storage Capacity**	**C3: Volume of Periodic Updates**
Scenario	Solution providers want to remain data owners and only share specific data via agreed interfaces.	Processing of machine data requires high computing power, which is not always available on the machine itself.	The high amount of generated machine data must be processed, aggregated and distributed.
Meta-Requirement	Data ownership and privacy must be guaranteed at all times.	The computing power must be sufficient for the chosen purpose.	Data processing and distribution must be carried out even when there is a high volume of data.
Design Principle	Data that resides outside of the DT and is worth protecting remains encapsulated - access only via defined interfaces.	Adequately equipped subsystems based on computing power or storage capacity needs.	Use a scalable data processing and data transmission setup.
		Federated Digital Twin	
Design Decision	Local data service that is connected to the DT using the Customer-Supplier pattern.	DT parts access and are accessed by distributed services/subsystems.	Message Queuing / Producer-Consumer framework for data processing and transmission.

Fig. 2: Overview of applied design principles induction

4.4 Formalizing Decisions Using the Decision Model and Notation (DMN)

By formalizing the decision-making using the DMN standard [Ob21], we address RQ3: Based on the identified design decisions, a Decision Requirements Diagram (DRD) is elaborated that shows the relationships between the decisions to be taken into account when developing an FDT. Starting with the originally conceived concerns, the connections, influences, and interdependencies between decisions are visualized in the DRD. Decision tables represent details about these decisions.

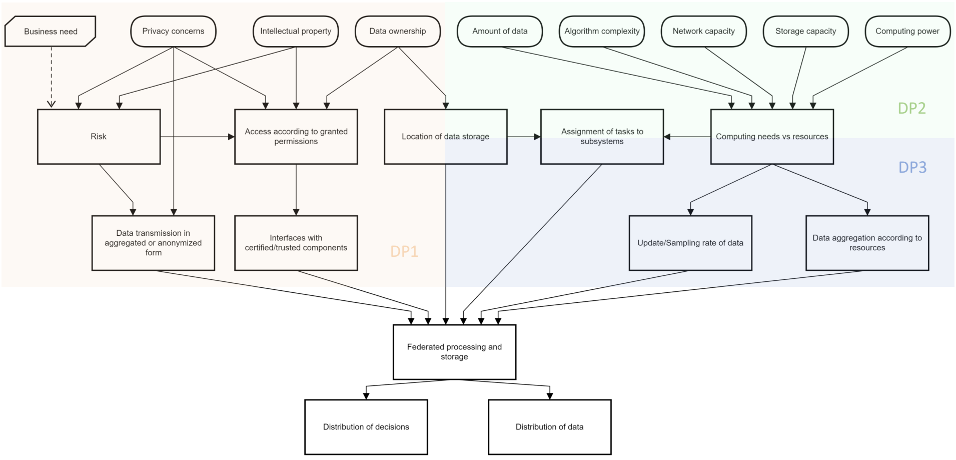


Fig. 3: Decision model represented as Decision Requirements Diagram (DRD)

For the notation, we chose DMN since it is an open standard for decision rules providing extensive specification and having wide acceptance, thus, offering the potential for generalization and transfer to other domains. One of the strengths of the DMN is the ability to show dependencies between decisions. DMN reduces complexity by decoupling decision and control flow logic [Fi18], i.e., decisions do not have to be depicted using process models. With the originally conceived goals and constraints (see Sect. 1), denoted as *DMN Input Data*, and the derived design decisions (shown as *DMN Decisions*), we created an exemplary DRD. Thus, we show the connection between decisions regarding the development of an FDT, and we give an example of how to use DMN for decision-making purposes for FDTs. According to the previous statements, *privacy concerns*, *intellectual property*, and *data ownership* (as input) can be assigned to DP1, thus influencing decisions concerning, e.g., data exposure (access, interfaces, ...). *Amount of data*, *algorithm complexity*, *network or storage capacity*, and *computing power* are inputs for decisions related to the trade-off between computing needs and computing resources. Both DP2 and DP3 derive from such considerations. Fig. 3 shows the resulting DRD.

Ultimately, all decisions affect *federated computing and storage* and potentially lead to the distribution of decisions and data. The DRD shows the interdependencies of the decision-making process very well. For example, if we consider *data ownership* as a goal that influences the decision regarding the *location of data storage*, which itself references the decision about the *assignment of tasks to subsystems* in the DT. At the same time, the latter could also be influenced by a decision about *computing needs vs. resources*.

When focusing on specific goals, we propose using *DMN Decision Services*. Even if human experts perform our decision-making, Decision Services are suitable for encapsulating decisions. For example, let us consider the case of data allocation (i.e., data access and data distribution). It could be designed in the way as shown in Fig. 4: there are two decisions, of which the first is about the data access policy. The inputs for this decision are the data owner's *privacy concerns* denoted as DMN input. Furthermore, there are several stakeholders in competition with each other. The decision about data access directly influences the decision about data separation. When deciding about data separation, we could also consider existing knowledge about the design of distributed systems. This knowledge is presented as tactics by [BCK21], represented as *DMN Business Knowledge*). The use of decision services can be applied for each goal, e.g., related to ownership/privacy, resources (computing power, storage space, network capacity), or availability (e.g., redundancy).

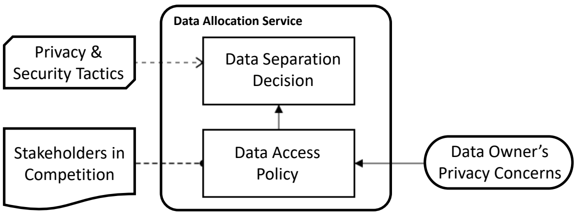


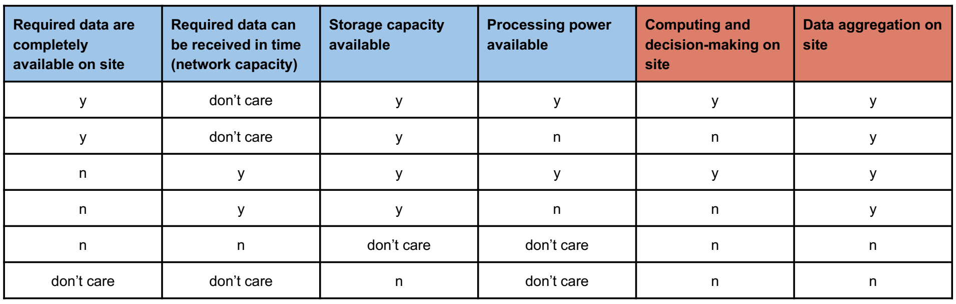
Fig. 4: Privacy decision as DRD

For representing decisions in detail, decision tables proved to be an appropriate way of formalization. As DMN also supports them, we propose decision tables to denote specific decisions in case of conflicts and competing goals. As an example, we examine the case of a plant operator (PO) considering providing data access for ADAM (see Fig. 5). The decision table represents the negotiation of a PO with the ADAM consortium. The input values (blue) are the goals of the negotiating partner, while the output values (red) contain options for action.

PO privacy risk	**acceptable delay for ADAM**	**data history needed by ADAM**	**data access granted by PO**	**measures due to PO's privacy concerns**	**measures for data accuracy in ADAM**	**arguments in negotiation, comments on decision**
high	none - required for immediate response	no	Push after demand by ADAM or triggered by PO	only for immediate evaluation by triggers or rules, no storage within ADAM	none	PO needs ADAM's urgent response
high	some delay accepted - no decision without access	no	Push as cyclic transmission, max delay = cycle time	no history is kept	Caching of most recent value	PO gets better results or decisions when new data is provided
medium	helpful for better decision - in case of missing access calculation with recent data	yes	Push as cyclic transmission	none	Caching of most recent value	
low	don't care	don't care	Pull	not relevant	not defined	Pull access is preferred by ADAM if granted by PO

Fig. 5: Data access decision table: ownership concerns vs. ADAM's data needs

According to DMN, decisions, as well as business knowledge, can be expressed using decision tables. As described above, such knowledge could be, e.g., existing knowledge about the decomposition and distribution of systems. For example, we consider a decision about computing at the current location (on-site). In this case, we could rely on (business) knowledge of resource-driven decomposition, i.e., the decision if a computation is executed on-site is dependent on the resources available; see Fig. 6.



Required data are completely available on site	**Required data can be received in time (network capacity)**	**Storage capacity available**	**Processing power available**	**Computing and decision-making on site**	**Data aggregation on site**
y	don't care	y	y	y	y
y	don't care	y	n	n	y
n	y	y	y	y	y
n	y	y	n	n	y
n	n	don't care	don't care	n	n
don't care	don't care	n	don't care	n	n

Fig. 6: Decision table: business knowledge on resource-driven decomposition

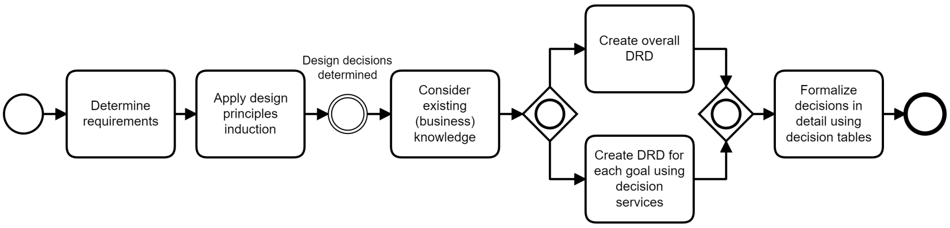


Fig. 7: Decision-making process

In summary, the application of the proposed approach for decision-making about FDTs, especially for other projects and use cases, can be realized as follows (also see Fig. 7):

1. Determine requirements and goals to be considered
2. Apply design principles induction to determine the resulting decisions
3. Consider existing knowledge (to not reinvent the wheel)
4. Create DRD as
    a) Overall DRD based on design principles (see Fig. 3) or
    b) DRD for each decision, use decision services if applicable (see Fig. 4)
5. Formalize decisions and business knowledge in detail using decision tables (see Fig. 5 and Fig. 6)

5 Results and Evaluation

Evaluation of Design Science artifacts is an essential part of DSR [GH13; Pe12]. As mentioned in Sect. 4.1, an evaluation in a real-world context is required to assess our approach, i.e., an evaluation in a real environment is mandatory—also known as *naturalistic evaluation* [VPB12]. The practitioners involved in the ADAM project formed a *focus group* to examine the perception of experts in the field [Bu10]. To evaluate our approach, we applied it to the already existing implementation in the ADAM project. In the following, we describe how the design decisions toward an FDT, based on the previously presented approach, affected the implementation of the ADAM project. The overall architecture resulting from the application of the decision process comprises the units given in Tab. 1: the machine and production system, the plant control station, the ADAM agent, and solution libraries (DD4).

The individual **machines** provide their configuration and runtime data from the DT via the MQTT protocol. To accumulate runtime data over a period of time, a so-called data aggregator for the **production system** has been implemented in an integrated time-series database InfluxDB. The aggregator provides average data over a timeframe but withholds more privacy-critical event data. In the **plant control station**, these data are accepted and checked against triggers for adaptation (DD1) by a rule engine. It uses rules to determine if the described circumstances occur (DD7). If rules apply, this is passed to the **ADAM agent**, thus withholding the actual machine data (DD3). The plant operator continuously controls the machine data and only provides abstract and summarized data to the ADAM agent (DD8). As a result, the amount and frequency of data transfer are reduced, and data computing and storage are distributed (DD5).

In the **ADAM agent**, an adaption, as well as a monitoring component, is defined as subcomponents that communicate via Apache Kafka. The latter was introduced because of the amount and frequency of data (DD6). In the adaptation component, possible adaptations to the trigger are compiled. A detailed description of the adaptation and adjustments—a ‘recipe’—is used for this purpose.

Furthermore, verified **solution libraries** from different drive (solution) providers can be addressed to retrieve updated component information suitable for the specific production system (DD2). These libraries are located at the drive (solution) providers because they contain sensitive data that must be protected and considered the providers’ intellectual property. Possible simulation and a knowledge-based monitoring component in the agent can warn against unnecessary adaptations. Proposed adaptations are presented to the plant operator, who might select the most suitable one. Because of the risk of interventions in the production process, downtime, and costs, decisions have to be approved by a responsible person. After the adaptation is approved and implemented, the machine configuration is updated in the DT.

The resulting prototype implementation has successfully identified adaptation scenarios using information from multiple parts of the FDT. It has determined suitable solutions (e.g., changes in software configuration, hardware replacement) using the solution libraries. Additionally, the experts have evaluated the approach and the solution: The decision process was comprehensible and the result feasible. Furthermore, applying a rigorous decision process leads to increased trust in the result and better acceptance of connecting the systems and sharing sensible data.

We conclude that the DDs applied for splitting and distributing the DT are suitable and thus answer **RQ1** (Which criteria are relevant to split a DT?) and **RQ2** (How can the DT parts be assigned to system parts?). Moreover, we presented the decision process using these criteria in response to **RQ3** (What is a general (formalized) decision process to distribute a DT?).

Finally, we relate the results to the overall goals from Sect. 1: The application in a real-world case study with a high level of complexity and the evaluation with experts have shown the approach to be appropriate for the given context, to provide *clarity* and *understandability*, and to *master complexity*. Furthermore, by providing a general DM and using a standard notation, the approach offers variability allowing *adaptation* in other use cases as well as *transferability* to other domains. In particular, it may be applied with different structures and domains, changed priorities, and design principles.

6 Conclusion and Future Work

In this paper, we presented a DM for splitting and distributing the DT. We evaluated the approach in the ADAM project, which is relevant for cases when a DT cannot be built as one single element in one place, e.g., due to insufficient resources or missing data availability. Therefore, the DM supports splitting a DT into an FDT and assigning FDT parts to the different computing units. Furthermore, the DM considers the issues and constraints regarding distribution, such as resources, privacy, availability, and similar. The DM has been developed following the DSR methodology and uses the standard DMN as notation. The DM and its application have been illustrated with a case study on factory automation. Practitioners evaluated it in its application in an industrial project. Data ownership issues and intellectual property and privacy concerns turned out to be of particular relevance in this project, and they required great attention in elicitation and decision-making.

In future work, we plan to research how extensions of a distributed DT can be performed that lead to meta-model changes, i.e., considering added parameters for a component. Synchronization of partly overlapping decision models constitutes another critical issue. Furthermore, research is necessary regarding optimal decisions that demand global optimization. Especially the optimization in the case of distributed data is an unsolved question. Moreover, reusable solutions within the solution portfolio need to be improved and optimized by learning from results, which demands verification of the improved solutions.

Acknowledgements

This research was partly funded by the German Federal Ministry of Education and Research (BMBF) within the project *ADAM – Autonomously aDApting Machines* (program *IKT 2020*, reference number *01IS18077A*).

References

[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.

[BCK21] Bass, L.; Clements, P. C.; Kazman, R.: Software Architecture in Practice. Fourth Edition, Addison-Wesley Professional, 2021.

[Be21] Bergs, T.; Gierlings, S.; Auerbach, T.; Klink, A.; Schraknepper, D.; Augspurger, T.: The Concept of Digital Twin and Digital Shadow in Manufacturing. Procedia CIRP 101/, 9th CIRP Conference on High Performance Cutting, pp. 81–84, 2021, ISSN: 2212-8271, URL: `https://www.sciencedirect.com/science/article/pii/S2212827121006612`.

[Bu10] Burgess, S.: The Use of Focus Groups in Information Systems Research. International Journal of Interdisciplinary Social Sciences 5/, pp. 57–68, Jan. 2010.

[DI16] DIN SPEC 91345: Reference Architecture Model Industrie 4.0 (RAMI4.0), Berlin, Germany: DIN Deutsches Institut für Normung e. V., 2016.

[Er22] Eramo, R.; Bordeleau, F.; Combemale, B.; Brand, M. v. d.; Wimmer, M.; Wortmann, A.: Conceptualizing Digital Twins. IEEE Software 39/2, pp. 39–46, 2022.

[Ev04] Evans, E.: Domain-driven design: tackling complexity in the heart of software. Addison-Wesley Professional, 2004.

[Fi18] Figl, K.; Mendling, J.; Tokdemir, G.; Vanthienen, J.: What we know and what we do not know about DMN. Enterprise Modelling and Information Systems Architectures 13/2, pp. 1–16, 2018.

[GH13] Gregor, S.; Hevner, A. R.: Positioning and Presenting Design Science Research for Maximum Impact. Management Information Systems Quarterly 37/2, pp. 337–356, 2013.

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

[He04] Hevner, A. R.; March, S. T.; Park, J.; Ram, S.: Design Science in Information Systems Research. Management Information Systems Quarterly 28/1, pp. 75–105, 2004.

[HG09] von Halle, B.; Goldberg, L.: The Decision Model – A Business Logic Framework Linking Business and Technology. Auerbach Publications, 2009.

[IB22] IBM: Operational Decision Manager, online, 2022, URL: `https://www.ibm.com/products/operational-decision-manager`.

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

[KMM20] Kaur, M. J.; Mishra, V. P.; Maheshwari, P.: The Convergence of Digital Twin, IoT, and Machine Learning: Transforming Data into Action. In (Farsi, M.; Daneshkhah, A.; Hosseinian-Far, A.; Jahankhani, H., eds.): Digital Twin Technologies and Smart Cities. Springer International Publishing, Cham, pp. 3–17, 2020, ISBN: 978-3-030-18732-3, URL: `https://doi.org/10.1007/978-3-030-18732-3_1`.

[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/, pp. 613–636, 2016.

[NJ82] Naumann, J. D.; Jenkins, A. M.: Prototyping: The New Paradigm for Systems Development. Management Information Systems Quarterly 6/3, pp. 29–44, 1982.

[Ob21] Object Management Group: Decision Model and Notation (DMN), Version 1.3, 2021, URL: `https://www.omg.org/spec/DMN`.

[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 20/1, pp. 7–10, 2011.

[Pe12] Peffers, K.; Rothenberger, M.; Tuunanen, T.; Vaezi, R.: Design Science Research Evaluation. In (Peffers, K.; Rothenberger, M.; Kuechler, B., eds.): Design Science Research in Information Systems. Advances in Theory and Practice. Springer, pp. 398–410, 2012.

[Pl18] Plattform Industrie 4.0: Reference Architectural Model Industrie 4.0 (RAMI4.0) – An Introduction, 2018, URL: `https://www.plattform-i40.de/PI40/Redaktion/EN/Downloads/Publikation/rami40-an-introduction.pdf`.

[Sc16] Schroeder, G. N.; Steinmetz, C.; Pereira, C. E.; Espindola, D. B.: Digital Twin Data Modeling with AutomationML and a Communication Methodology for Data Exchange. IFAC-PapersOnLine 49/30, 4th IFAC Symposium on Telematics Applications TA 2016, pp. 12–17, 2016, ISSN: 2405-8963, URL: `https://www.sciencedirect.com/science/article/pii/S2405896316325538`.

[Sc17] Schleich, B.; Anwer, N.; Mathieu, L.; Wartzack, S.: Shaping the digital twin for design and production engineering. CIRP Annals 66/1, pp. 141–144, 2017, ISSN: 0007-8506, URL: `https://www.sciencedirect.com/science/article/pii/S0007850617300409`.

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

[Ta19] Tao, F.; Zhang, H.; Liu, A.; Nee, A. Y. C.: Digital Twin in Industry: State-of-the-Art. IEEE Transactions on Industrial Informatics 15/4, pp. 2405–2415, 2019.

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

[WD20] Wright, L.; Davidson, S.: How to tell the difference between a model and a digital twin. Advanced Modeling and Simulation in Engineering Sciences 7/, 2020.