Tackling Complexity in Smart Manufacturing with Advanced Manufacturing Process Management

Tackling Complexity in Smart Manufacturing with Advanced Manufacturing Process Management

Konstantinos Traganos

SIKS Dissertation Series No. 2022-31

The research presented in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

It was part of the HORSE project and received funding from the European Union's Horizon 2020 research and innovation program under grant agreement no. 680734. It was also part of the OEDIPUS project, funded by EIT Digital. Moreover, it was partially supported by the SHOP4CF Project funded by the European Union's Horizon 2020 research and innovation program under grant agreement no. 873087.

A catalogue record is available from the Eindhoven University of Technology Library.

ISBN: 978-90-386-5583-3

Printed by: Proefschriftmaken | | www.proefschriftmaken.nl
Cover design by: Stefanie van den Herik

Tackling Complexity
in Smart Manufacturing
with Advanced
Manufacturing Process Management

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Technische Universiteit Eindhoven,
op gezag van de rector magnificus prof.dr.ir. F.P.T. Baaijens,
voor een commissie aangewezen door het College voor Promoties, in het
openbaar te verdedigen op woensdag 09 november 2022 om 13:30 uur

door

Konstantinos Traganos

geboren te Lamia, Griekenland

Dit proefschrift is goedgekeurd door de promotor en copromotor, en de samenstelling van de promotiecommissie is als volgt:

voorzitter:	prof.dr.ir. G.J. van Houtum
promotor:	prof.dr.ir. P.W.P.J. Grefen
co-promotor:	dr.ir. I.T.P. Vanderfeesten
leden:	prof.dr. M. Reichert (University of Ulm)
	prof.dr. M.G.J. van den Brand
	prof.dr. D. Karastoyanova (Rijksuniversiteit Groningen)
	prof.dr. J. van Hillegersberg (Universiteit Twente)
	dr. O. Türetken

Het onderzoek of ontwerp dat in dit proefschrift wordt beschreven is uitgevoerd in overeenstemming met de TU/e Gedragscode Wetenschapsbeoefening.

Summary

The evolution of the manufacturing paradigm has always been characterized by both market trends and technology advancements. In the ongoing 4th industrial revolution, customers demand more individual products, of high-quality and, if possible, with same-day delivery. The versatility of products and the configuration options upon ordering give rise to mass-customization and personalization. In the context of a global business environment, in which the competition is extremely high and the changes can be disruptive, this market shift puts pressure on manufacturers who seek for flexibility to satisfy the demands. They have to adapt and reconfigure their systems in order to offer the product variety within short lead times and be responsive to changes. Typically, product variety introduces production variety that often comes with a high degree of complexity. The recent technology developments offer more flexibility in production operations. Versatile robots can perform various operations. Collaborative robots (cobots) increase efficiency by allowing robots and human operators to work together. Augmented Reality (AR) systems support operators in their daily tasks, which are getting more complex. Automated guided vehicles (AGV) transport material and products around a factory, without human intervention. Smart sensors gather any kind of values from devices that help in predictive maintenance or decision making. And all these developments are leveraged by the connectivity that the Internet-of-Things (IoT) and cloud computing provide. However, robotic solutions are often employed in disparate work cells, following a vertical orientation in their robot control processes. This usually leads to isolated, fragmented developments that do not solve the need for production adaptability and flexibility at the level of the entire process. We see, thus, that current production environments are getting complex and the transition from a traditional factory into one that hosts the new smart technologies is challenging.

The research presented in this thesis extends previous research on the application of theories, techniques and tools from the well-developed business process management (BPM) paradigm in smart manufacturing, aiming at tackling operations complexity. BPM can offer flexible ways to design and configure operations, methods to respond to the unexpected changes and events, dynamic resource allocation mechanisms and drive integration of systems for better transparency and process efficiency. As the paradigm has been mostly applied in business sectors, where information processing is dominant, adaptations and extensions are required for application in the manufacturing domain, in which physical aspects are involved. By performing design science research, knowledge, in the form of artefacts, is generated in order to help practitioners for utilizing BPM in complex smart production environments. This thesis focuses on the following four main artefacts: 1) modeling patterns and mechanisms on representing complex production processes, 2) a categorization of exceptions and corresponding handling strategies, 3) the specification of a manufacturing process management system (MPMS) to support the modeling and execution of end-to-end manufacturing processes, and 4) an architecture model of an advanced MPMS, as a blueprint to realize an information system that enables horizontal process integration and

direct process control for vertical integration. The advanced MPMS architecture incorporates the conceptual designs of the first three artefacts. This research complements existing work on application of BPM in smart manufacturing, which mainly focuses on dynamic resource allocation. The contributions presented in this thesis are the main ingredients of an advanced process management approach to overcome production complexity and enable smoother introduction of smart technologies.

Realizations of the designed artefacts are demonstrated at various production enterprises across Europe, within three European research and development projects, proving application feasibility and gaining practical insights into the implementation and ease of use of the solutions.

Preface

The past few years as a Ph.D. candidate, culminated in this dissertation, was an invaluable journey from many perspectives. I can now see myself how I developed as both a researcher and a person through the hard work and its outcomes, the motivation to generate applied and useful knowledge, the collaboration with colleagues but also the times of solitude, the perseverance and patience to overcome challenges and struggles. Of course, what is presented in this thesis and its underlying work, would not exist without the contribution and support from colleagues, friends and family. Thus, I use the opportunity to express my warmest gratitude to the people involved in my Ph.D. trajectory.

I would like to start with the supervision team and the colleagues who contributed the most directly to my work. Dear Paul, I feel blessed to have you as my promotor. You methodically guided me through the process, helped me structuring the concepts of my research, especially when I was struggling to see the bigger picture, and provided sharp feedback to tackle open issues. Working with you since my master studies, you are more than my promotor. You have been mentoring me on how to nicely design information systems architectures, how to see the "why" behind the "what" and "how" of building solutions, and how to clearly communicate my work in academic settings. This mentorship has shaped me as a researcher and engineer. Knowing you all these past years, we also had the chance to have more personal discussions through our nice lunches, dinners and business trips we shared, which I greatly enjoyed and learned from. Thanks for everything!

Dear Irene, as my co-promotor and daily supervisor, you provided all the settings with the acquired large projects to apply our research ideas. You also gave me the freedom and space to manoeuvre and find my own research interests. Your "helicopter view" approach and feedback enabled me to see the greater value of my work when I was drowning with small details. Your experience on how to convey "into the point" information that will be easy for the reader to understand was valuable for improving my writing skills. Moreover, you always encouraged and supported me with our open discussions during my stressful periods. Thanks for our fruitful collaboration!

I want to thank my ex-colleague Jonnro. The significant work he has done in the projects, resulting in a well-written Ph.D, thesis, together with the interesting discussions, even after he left the team, were valuable inputs and a useful guide for me to write the current thesis. In addition, I want to thank Dillan and John for they hard work during their master projects. Moreover, I want to thank Zuzanna for handling all organizational work in the last project, giving me time to focus on my thesis.

Next, I want to thank all the colleagues with whom I collaborated within the European research projects. This collaboration gave me the opportunity to apply my ideas and work with people with different professional backgrounds and thus, getting valuable experience on how my work fits in a broader context. I am extra grateful to those who devoted their time to help me with the interviews and evaluation sessions. A special thanks to Ruud Keulen for his openness on applying our research ideas in his organization. I am also grateful to Tasos

Georgakopoulos who supported the technical developments in the last project, making the finalization of my work possible.

Moving to my broader research circle, I would like to thank all the colleagues of the Information Systems research group who created a nice environment to work, to exchange ideas and share interesting stories during our social events. In particular, I would like to thank (in alphabetical order) Amirreza, Bilge, Caro, Ege, Jason, Paulo, Reza, Rick, Sander, Shaya, Sud and Zeynep for the nice time we spent together all these past years. Moreover, I would like to thank Emmy for solving any practical needs at the office.

As for the people closest to me, outside my research circle, I want to express my greatest gratitude to my parents, Christos and Eleni, for their unconditional love, support and sacrifices. You have devoted most of your life to provide me and my sister with the best opportunities to thrive. This thesis is devoted to you, as you deserve to be proud of this achievement. To my sister, Stella, thank you for being there always for me and keeping a true sibling bond.

Lastly, I would like to express my special gratitude to Liliāna for her unwavering support, motivation and patience the past couple of years. You already know the basic concepts and the structure of my thesis even without first having read a single line. With you by me, I was able to stay strong throughout the tough process of writing this thesis. As the lyrics of a Latvian song say: *"Mākslas darbi rodas mokās (Artworks are born/being created in suffering)"*. I hope this thesis is also rewarding for you.

As for the reader, I would like to thank you upfront for your time to read hopefully the entire thesis. I hope you find interesting and valuable either the general concepts behind this research or even the small details. In the end, I will be content if my work, in some form, contributes to a smooth embracement of Industry 4.0 technologies in smart manufacturing.

Konstantinos Traganos

Eindhoven, August 2022

Table of Contents

Table of Figures

Table of Tables

CHAPTER 1

Introduction

This first chapter introduces the research presented in this thesis. We start in Section 1.1 with background information on the history of manufacturing that shows its evolution, and further discuss, in Section 1.2, the general context in which the research is conducted. In Section 1.3, we elaborate on the identified problem and in Section 1.4 we present our proposition to address it. Having delineated, in Section 1.5, the research scope within which the problem is tackled, we state in Section 1.6 the research objectives. The approach and the methods that we use to reach those objectives are discussed in Section 1.7. Finally, Section 1.8 outlines the structure of this thesis.

1.1 Brief history of manufacturing

Industrial manufacturing is evolving through time, characterized by powerful changes termed as "revolutions", which have even shaped the world history (Stearns, 2020). Each of them is described by dominant production paradigms, emerged from society and market needs, technology and process enablers.

The First Industrial Revolution started at the end of the 18th century, with the introduction of machines driven by water or steam power to manufacture products. The era is characterized by *craft production* (or customer production (Wang et al., 2017)), as the users first set the requirements for the products, which were then designed and made by the craft producer. Products were made at a limited number and at high costs. The textile industry was one of the first domains to embrace mechanized manufacturing methods.

The Second Industrial Revolution occurred in the early 20th century, marked by the invention of the moving assembly line by Henry Ford in 1913. Division of labor and standardization enabled the *mass production* of identical products[1], in high quantities, at low costs but in limited variety.

The Third Industrial Revolution was introduced in 1970s-1980s as a response to market needs for more diversified products. With the adoption of digital and automation technology into manufacturing, producers were able to design and create a large variety of products, leading to the *mass customization production* paradigm (or flexible production (Jovane et al., 2003)). Industrial robots, computer integrated systems and enterprise information systems featured higher productivity at relatively low costs.

The Fourth Industrial Revolution is an ongoing phase, started at the beginning of 21st century, with the extended use of integrated digital technology. More and more devices are connected to the internet, creating a networked environment and coupling physical systems to digital ones, leading to cyber-physical systems. The driving force is the customer demand for even larger product variety and more personalized products. The globalization, by creating a

[1] H. Ford had remarkably stated "*Any customer can have a car painted any color that he wants so long as it is black.*" (Ford, 2019).

single, worldwide market, offers the opportunity to satisfy this demand and gives rise to the *mass personalization production* paradigm.

Figure 1 visualizes the evolution of the production paradigm, viewed from different dimensions.

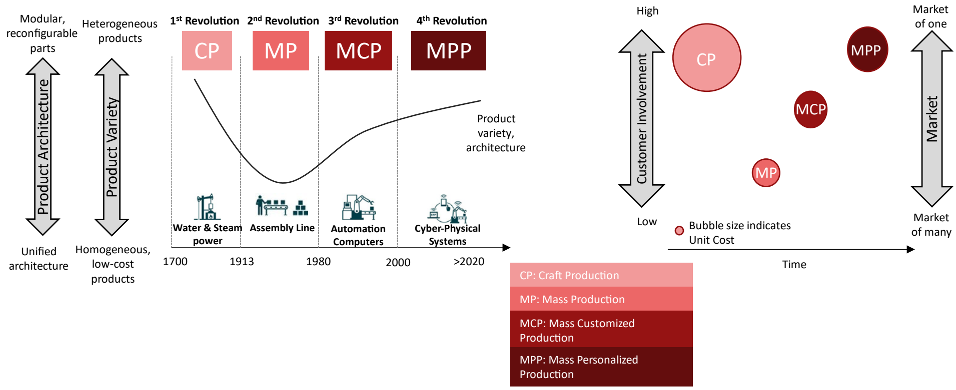


Figure 1: Evolution of production paradigm in different dimensions (inspired by (Wang et al., 2017) and (Jovane et al., 2003)).

Towards the mass personalization, we encounter a shift from a traditional manufacturing-based approach where value is created for the customer, to a service approach where value is created with the customer as a collaborative partner (Salunke et al., 2011). This value co-creation in a service-dominant logic (Grefen, 2015; Kowalkowski, 2011); (Vargo & Lusch, 2004) requires manufacturers to embrace the customer in their operations. The changes are disruptive and we experience radical changes in business models, like for instance in the automotive industry where many car manufacturers[2] are shifting to online sales, giving the possibility to customers to customize and order directly the car they desire, even bypassing the car dealers[3].

Giving, though, the opportunity to customers to design and configure the product they want, be it big or small (Figure 2 shows two examples of online order configuration and customization systems), demands that the manufacturers have the required capabilities to do so. They should have the needed resources, the right technology and the ability to adapt their processes in order to satisfy the increasing customer demands for mass customization and personalization (Tseng & Piller, 2003) and keep up with or even be ahead of the competition.

---

[2] https://europe.autonews.com/automakers/why-vw-ford-volvo-others-are-accelerating-shift-online-sales-europe

[3] https://www.forbes.com/sites/larrylight/2020/11/02/personalization-will-change-your-car-dealership-experience-forever/

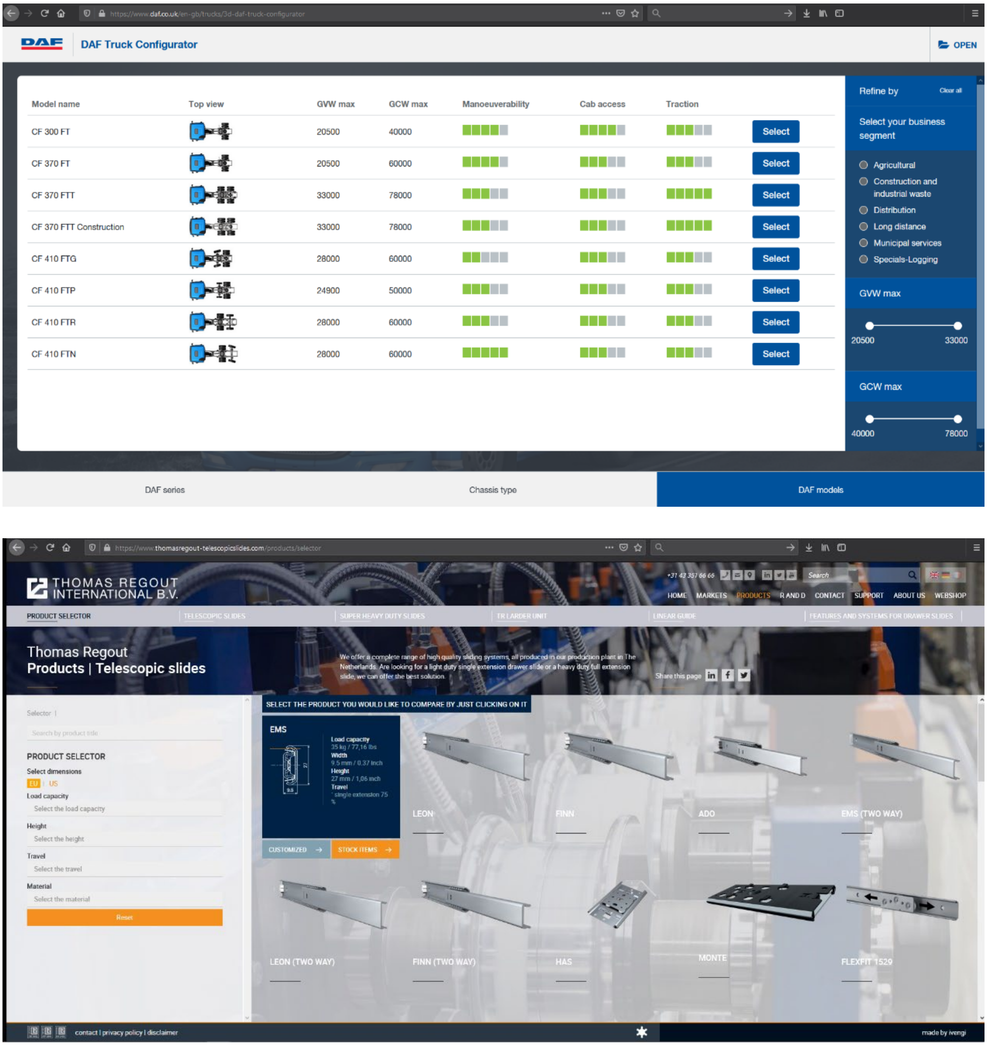


Figure 2: Examples of online product configuration/customization for direct ordering - trucks[4] *(top), telescopic slides*[5] *(bottom).*

The technological advancements of the ongoing industrial revolution offer manufacturers many opportunities but also pose many challenges (Khan & Turowski, 2016; Mosterman & Zander, 2016). This research aims to tackle a few of those challenges, from a few given perspectives.

1.2 Research context

Until the recent years, the market demands of each period of the manufacturing paradigm have well been addressed by the various types of manufacturing systems that have been

[4] https://www.daf.co.uk/en-gb/trucks/3d-daf-truck-configurator.
[5] https://www.thomasregout-telescopicslides.com/products/selector.

developed. Mass production at affordable costs has been achieved with dedicated manufacturing lines (DML)[6]. In such setups, material and products move through a transfer line, from one station to the next one to undergo the corresponding operations or treatments (Koren et al., 1999). These stations, aimed at handling high volumes, are rather fixed, with typically little configurability. While DML are cost effective, their rigidity does not allow for customization or production of a larger variety of products. These needs have been addressed by flexible manufacturing systems (FMS), consisting of general-purpose computer numerically controlled (CNC) machines and other programmable automation. Different products can be produced by the same system with varied volume and mix. However, equipment in FMS setups is typically expensive and with combination with the low throughput time (due to the single-tool operation), make the cost per part relatively high (Koren et al., 1999).

A new class of systems, called reconfigurable manufacturing systems (RMS), emerged to provide the versatility of machines for producing a wide range of products (Koren & Shpitalni, 2010). These systems make use of changeable tools that can be reconfigured per production run, offering in this way the capability of producing smaller batches of variable products. Apart from being modular, RMS are convertible, as individual modules can be repositioned/re-oriented on the machines, and scalable, as new machines can be relatively easily added in the production setup (Landers et al., 2001). RMS can achieve high throughput as DML systems, providing also the flexibility for customized production like in FMS. However, all these types of systems are not well-suited to support concepts that the new era of "Industry 4.0" brings, such as systems and technology interoperability and consciousness through intelligence, self-awareness and self-configuration (Qin et al., 2016).

Industry 4.0 is a term originally coined by the German Academy of Science and Engineering (acatech[7]) to describe the fourth stage of industrialization, in a national initiative to secure the future of the German manufacturing industry (Kagermann et al., 2013). Since then, the term has been widely used to largely denote the developments in manufacturing in the fourth industrial revolution. These developments are often described as "smart" or "intelligent" due to their advanced character and possibilities. The terms Industry 4.0 and Smart Manufacturing are often used interchangeably to describe similar concepts, however they are not strictly synonymous, as illustrated in Figure 3 (Brouns, 2019) (other terms such as smart city, smart mobility and smart health are used to describe developments in various domains in the current era but are not included under the Industry 4.0 term, which has a manufacturing perspective).

---

[6] A list of terms and abbreviations is available in Appendix A.
[7] https://en.acatech.de/

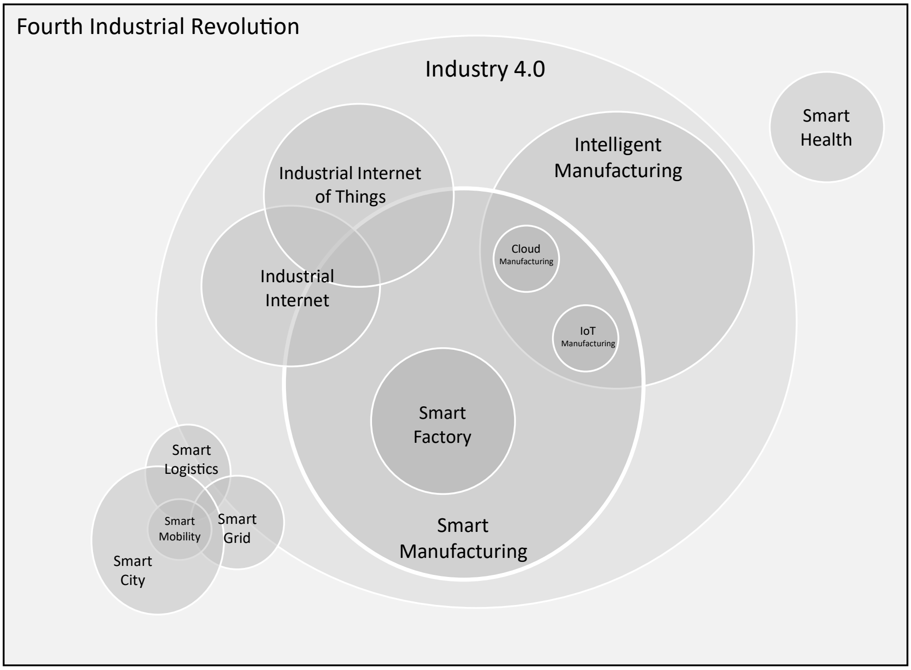


Figure 3: Related concepts in the fourth industrial revolution (from (Brouns, 2019)).

Apart from the German initiative and the term *Industry 4.0*, other countries have announced, in the past years, their strategies for the future of manufacturing; the *New Industrial France*[8] by France, the *National strategic plan for advanced manufacturing* (Holdren et al., 2012) and the *National Network for Manufacturing Innovation* (Molnar, 2015) by the United States of America, *Made in China 2025* (Wübbeke et al., 2016) by China, the *Industrial Value Chain*[9] by Japan. No matter though the differences in initiatives and the terms used, manufacturing is going through disruptive changes.

From a technology perspective, there are rapid developments on the equipment and techniques to manufacture products. Versatile robots, with the appropriate end effectors attached, can switch modes and perform various operations (Heyer, 2010). By programming by demonstration (Dillmann & Friedrich, 1996; Dey et al., 2004) manipulators can more easily add new functionalities to robots, making their utilization more efficient. Collaborative robots (cobots) increase efficiency by allowing robots and human operators to work together (Bejarano et al., 2019). Augmented reality (AR) systems support operators in their daily tasks, which are getting more complex (Khan et al., 2011; Longo et al., 2017). Automated guided vehicles (AGV) transport material and products around a factory, without human intervention (Le-Anh & de Koster, 2006), promising increased productivity (Fragapane et al., 2020). Smart sensors gather any kind of values from devices that help in predictive maintenance or decision making. And all these developments are leveraged by the

[8] https://www.economie.gouv.fr/files/files/PDF/industrie-du-futur_dp.pdf
[9] https://iv-i.org/wp-test/wp-content/uploads/2017/09/doc_161208_Industrial_Value_Chain_Reference_Architecture.pdf

connectivity that the Internet-of-Things (IoT) (Atzori et al., 2010) and cloud computing (Liu & Liu, 2010; Zhang et al., 2014) provide.

From a business perspective, there are new market pull forces, such as the increasing demand for customized and personalized products, with higher quality and, if possible, same day delivery. On a global level, business environments are getting more dynamic with a result of high fluctuations in demand for materials and products. Manufacturing enterprises strive to retain or increase their efficiency in operations and to deal with the production of small series of products, while on the same time they pursue the flexibility (Mishra et al., 2014) to quickly reposition themselves and reconfigure their competences (Tan & Wang, 2010) in order to stay competitive. All these market forces shift traditional supply chain models, which are typically based on manufacturing-to-stock approaches according to sales predictions (production-driven), into demand chain models, with a focus into value and outcome provisioning (Christopher & Ryals, 2014; Grefen et al., 2021). The implementation, though, of demand chain models require near-real time synchronization of manufacturing processes and their context.

The recent technologies that Industry 4.0 brings can enable manufacturers to respond to the current business requirements, driven by the market forces. There are promises for increased productivity, higher efficiency, flexibility and labor cost reduction (Dalenogare et al., 2018; Hofmann & Rüsch, 2017). However, the transition from a traditional factory into a smart one is a challenging endeavor, as the optimal utilization of the new technologies into manufacturing operations in complex and dynamic environments is not an easy task.

1.3 Problem identification

In the line of mass customization and personalization, manufacturers have to provide product variety which can have a great impact on operations performance (ElMaraghy et al., 2013; Johnsen & Hvam, 2019; Park & Okudan Kremer, 2015). Product variety imposes variety on production equipment and processes (Brunoe & Nielsen, 2016). The "high mix – low volume" production (i.e., high number of predefined product variants and low volume per variant) can cause complexity in operations (Hu et al., 2011) and often demands for fast equipment and tool changeovers. Reconfigurable machines and versatile robots are deployed to perform as many operations as possible and support the efficient production of smaller batches of various and customized products. But as product specifications and customer requirements are getting more and more sophisticated, production scenarios are getting more complex as well. Raw materials and (semi-finished) products typically have to go through a series of activities which involve various equipment and human resources. These activities may vary per batch or lot and their coordination is not an easy task. In case of small batch sizes and many product variants, the final assembly activities are often performed manually for better performance (Michalos et al., 2010). While the latest collaborative robots help human operators in production and assembly, by providing task dexterity, speed and quality, their utilization in production requires redesign of traditional workcells (Bruno & Antonelli, 2018). Moreover, the co-presence and collaboration of humans and robots require a lot of attention with respect to safety requirements to prevent hazards for humans (Reniers, 2017). Such safety requirements and restrictions can add extra complexity to current production scenarios.

Apart from the mass personalization trends, the manufacturing domain is currently characterized by fierce competition, a high degree of globalization and increased market uncertainty (Choi et al., 2016). Dynamic environments, which as described by Miller & Friesen (1983) are characterized by the rate of change (velocity/volatility) and its unpredictability, can directly affect the performance and sustainability of an enterprise (Nitsche & Straube, 2020; Saldanha et al., 2013). Rapid and unpredictable changes cause uncertainty, which can be encountered in various phases of manufacturing and supply chains in a broader perspective, ranging from the firm's environment down to the lowest task within the firm (Miller & Shamsie, 1999; Sawhney, 2006). According to Angkiriwang et al. (2014) uncertainty is classified in the supply chain context as upstream (supply) uncertainty, internal (process) uncertainty, and downstream (demand) uncertainty. Manufacturers might face material unavailability or late supplier's delivery. With the growing engagement of customers in their chain of activities, they might also face demand fluctuations, order specifications changes or last-minute cancellations. Such exceptional situations have a direct impact on internal business processes and manufacturing operations. In extreme cases, they might even have to completely alter their facilities and operations, as, for example, many car manufacturers did in the beginning of the COVID-19 pandemic[10], when they turned their vehicle production work-places into medical equipment ateliers for aiding on the excess demand on masks, respirators, ventilators, etc. Regarding the process uncertainty, the occurrence of unexpected events and the deviations from schedules are more likely to increase with the growing introduction and utilization of new technologies, as the probability of equipment and machinery malfunctions and failures is proportional to the count of resources.

The changes caused in dynamic environments require humans and machines to adapt and reconfigure their activities. However, the control systems of machines and robots are often not flexible enough to respond to changes and cope with resource relocation or alterations (Newman et al., 2008). Also, it is typically hard to transfer tasks from robotics to humans and vice versa, as each actor class is controlled differently and independently (Tsarouchi et al., 2016); robots and machines are forced to action through their control systems, while humans receive instructions orally, written, or visually through screens. Think, for example, a scenario of an AGV raising an alert of low battery capacity and a human operator having to take over a materials transportation task. With a rigid process design and reconfiguration, resources are under-utilized (Erasmus, Vanderfeesten, Traganos, Keulen, et al., 2020) and production operations can get complex if not properly managed.

Regardless of the high attention and the effort needed to embrace new technologies in operations, manufacturers should aim to include automated devices that will increase production efficiency and quality. However, as the acquisition is typically done in stages, it is very common that new robotic solutions are employed in disparate work cells, following a vertical orientation in their robot control processes. This normally leads to isolated, fragmented developments that do not solve the need for production adaptability and flexibility at entire process level. Moreover, as the number of systems and technologies increase, typically based on different control regimes and offered by multi-vendors (Weyer et al., 2015) their integration is a challenge (Dalmarco et al., 2019; Sanchez et al., 2020). As Kagermann et al. (2013) recommend, apart from vertical integration of manufacturing systems within the factory, horizontal integration of value networks and end-to-end digital

[10] https://www.caranddriver.com/news/g32041246/automakers-gowns-masks-ventilators-coronavirus/

integration of engineering across the entire value chain are key concepts of implementing the Industry 4.0 initiative. Cross-functional process integration (Brettel et al., 2011; Tang, 2010a) is crucial, but existing infrastructures are not ready to support it (da Xu et al., 2018). There exist well-developed information systems suitable for different types of functions (e.g., Enterprise Resource Planning (ERP), Manufacturing Execution Systems (MES) – presented in detail in Chapter 2), but their interoperability issues and the poor process alignment hinder flexibility.

The various factors described above lead to a general complexity issue, from operations perspective, in smart production environments, as illustrated with a cause-and-effect diagram (as proposed by (Ishikawa, 1990)) in Figure 4.

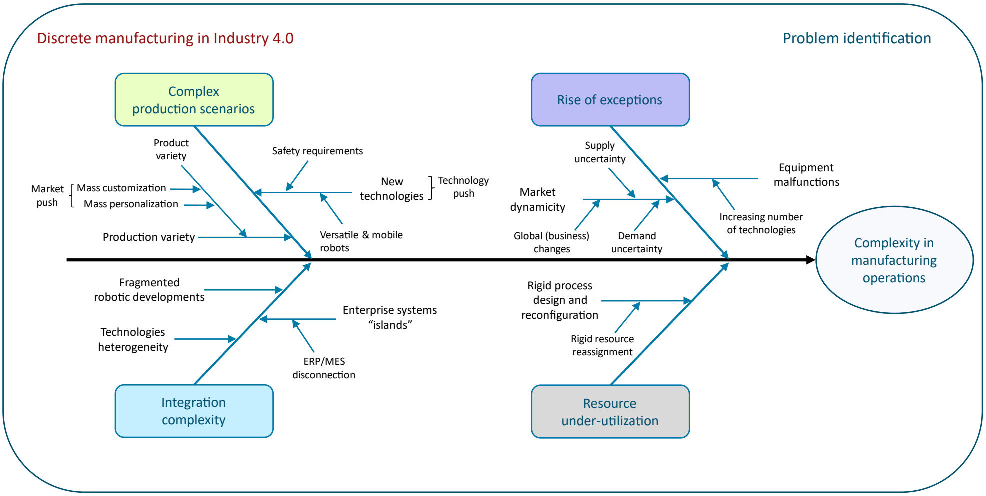


Figure 4: Problem identification presented in an Ishikawa (fishbone) diagram.

The identified problems and causes, which are discussed in more details in Section 2.1.6, are distilled into the following problem statement:

Problem statement

Production environments face an increased process complexity in their effort to enter the smart manufacturing era, which is characterized by a high degree of variety and dynamism.

1.4 Proposition

Operations complexity can pose difficulties and challenges to manufacturers to adopt new technologies and keep up with the competition. The various factors that lead to complexity, as discussed in the previous section, should be addressed towards tackling the identified problem. The production variety, caused by the product variety, requires robust process design (Salvador et al., 2009). Manufacturing activities should be modeled, so as the sequence of who performs what is clear. This is more crucial when the production scenarios are getting more complex, when there is an increasing number of new technologies and resources involved and when there is a high level of collaboration between humans and machines. The modeling of the processes should allow for synchronization of activities during execution, often in near real-time, for higher production efficiency. With the use of well-defined modeling patterns and mechanisms, process designers shall be able to represent the requested activities in the exact way that should allow for smooth execution. Moreover, processes, both business and manufacturing (the distinction of which we discuss in Section 1.5), should be modeled flexibly, i.e., with low sensitivity to changes (Chryssolouris et al., 2013). Especially in current dynamic environments, with high uncertainty, the need for responsiveness to exceptional events is imperative (Wang et al., 2014). That, of course, requires a structured classification of exceptions and proposed handling strategies. The classification should include events occurring both on business (e.g., last-minute order cancellations) and on operational levels (e.g., machine breakdowns). With respect to resource under-utilization, dynamic resource allocation mechanisms are required for selecting, also during runtime, the most suitable actor to perform a specific task. Lastly, the integration complexity issue shall be addressed by an information system (IS) able to provide both horizontal, cross-functional integration and vertical (for direct process control) integration. Such a system shall orchestrate end-to-end business and production processes performed by heterogeneous actors, coupling the "cyber" aspect of business information processing with the "physical" aspects of robotics and devices, as part of a cyber-physical system (CPS). Of course, as the developed system shall integrate and be integrated with other systems, it should be based on well-adopted standards (Lu et al., 2016; Weyer et al., 2015).

The propositions discussed above can be realized with concepts, methods and tooling from the business process management (BPM) paradigm, whose employment helps organizations to be more responsive to an increasingly changing environment (Lindsay et al., 2003). BPM, as defined by van der Aalst et al., (van der Aalst et al., 2003) is a paradigm for *supporting business processes using methods, techniques, and software to design, enact, control, and analyze operational processes involving humans, organizations, applications, documents and other sources of information*. A business process is defined as *the combination of a set of activities within an enterprise with a structure describing their logical order and dependence whose objective is to produce a desired result* (Aguilar-Savén, 2004). On a more philosophical level, *a process*, in general, *is a coordinated group of changes in the complexion of reality, an organized family of occurrences that are systematically linked to one another either casually or functionally* (Rescher, 1996). In that respect, manufacturers have to structure and coordinate their activities (either business or manufacturing), which can get more and more complex, in order to achieve their corporate objectives.

This research advocates BPM as a good candidate approach towards tackling complexity in manufacturing operations, as it can support all the aforementioned aspects that are proposed as solutions to the various factors causing complexity. With respect to modeling, the paradigm offers well-defined notations, such as the Business Process Model & Notation 2.0[11] (BPMN 2.0) or Petri Nets (van der Aalst, 2009), as graphical representations that provide a comprehensive and common understanding of a business process. The notations typically have a formal foundation to avoid ambiguity (Aalst, 1998) and enable process analysis (Aguilar-Savén, 2004; Wodtke & Weikum, 1997). Regarding responsiveness to exceptions and changes, various handling patterns have been developed in the context of business process management (Rinderle & Reichert, 2006; Russell et al., 2006a). With respect to resource and task allocation, BPM provides different strategies to select the right resources to perform a task (Dumas et al., 2018). Finally, as all the business processes have to be controlled and enacted in an automated way, a Business Process Management System

[11] https://www.omg.org/spec/BPMN/2.0/

(BPMS) typically provides the support of executing the defined process models, handling exceptions during runtime, delegating tasks to resources based on the allocation strategies and monitoring the running process instances. As the processes in concern might involve various resources and information systems, a BPMS improves enterprise integration by enabling the invocation of applications and services across heterogeneous systems (Harmon, 2010; van der Aalst, 2013). To guarantee this integration, BPM systems are often designed and implemented according to well-established information systems reference architectures, such as are the Workflow Reference Model (Hollingsworth, 1995) and the Mercurius Reference Architecture (Grefen & Remmerts De Vries, 1998).

BPM has originated from business sectors where information processing is dominant, e.g., finance (Brahe, 2007), but it has also been extensively applied in healthcare (Reichert, 2011; van Gorp et al., 2013), automotive (Grefen, Mehandjiev, et al., 2009) and transportation (Baumgraß et al., 2015), where physical entities are included as well. In this research, the terms Manufacturing Process Management (MPM) and Manufacturing Process Management System (MPMS) are used often to denote the application of the BPM paradigm in the manufacturing domain. In that respect, the proposition of the current research can be summarized in the following sentence and illustrated in more details in Figure 5:

Proposition

Process management theories and techniques applied in the manufacturing domain can tackle the process complexity in smart production environments.

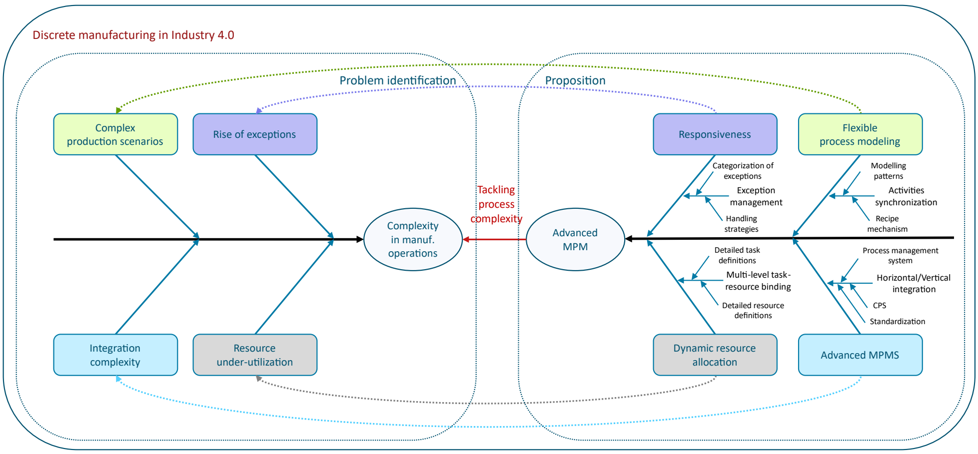


Figure 5: Overview of proposed solutions to tackle the identified issues in discrete manufacturing.

BPM in manufacturing is ongoing research (Janiesch et al., 2017) (as discussed extensively in Section 2.1.7.4. Erasmus' work (2019) on application of BPM in discrete manufacturing has proven feasibility and distinct advantages. That work focuses on main aspects such as modeling manufacturing operations as business processes and providing a system (MPMS) to enact these models. Its main focus is on resource allocation, by providing an algorithm to select the most appropriate (in terms of various criteria) actor to execute a task during runtime. While this covers the resource under-utilization issue, that work has to be extended with functionality that will enhance and cover all the other identified issues in manufacturing.

More specifically, the current research proposes extensions on three aspects: i) design of more flexible patterns to model the complex manufacturing processes, ii) support exception handling with BPM approaches, and iii) extend the scope of MPMS to offer integration functionality with other enterprise systems. These extensions give the notion of the "advanced" solutions (as denoted in Figure 5) and are illustrated, with respect to Erasmus' work (2019), in Figure 6.

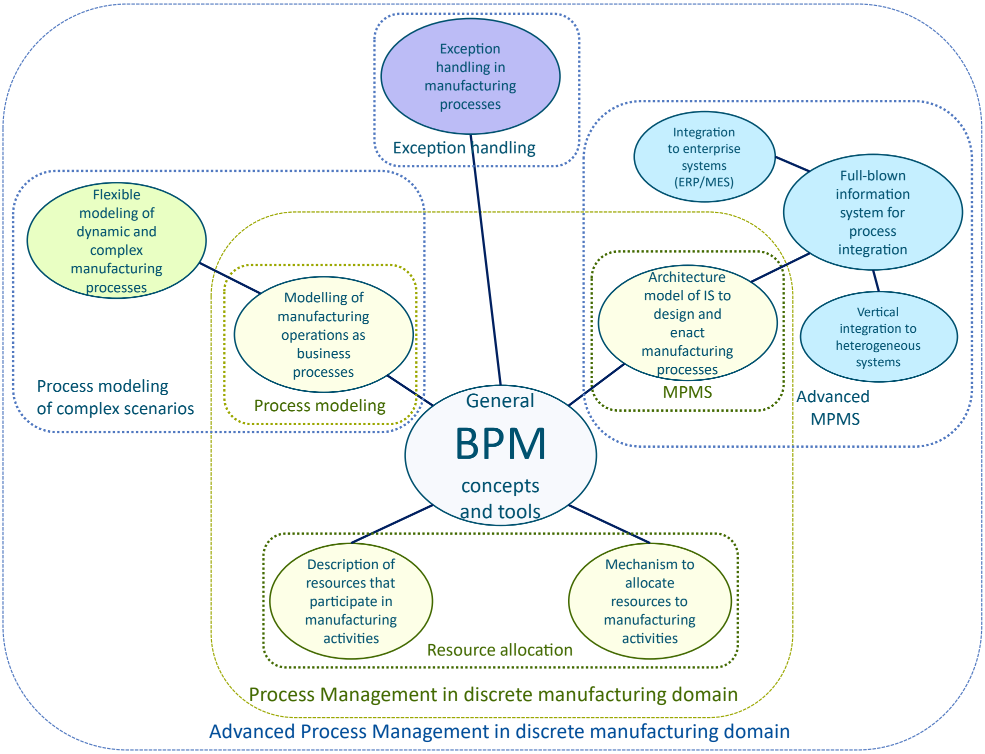


Figure 6: BPM concepts as proposition to tackle the process complexity in smart manufacturing. Concepts in green dashed box has been covered by previous research (Erasmus, 2019). Concepts outside the green box (and within the blue dashed box) are covered by the current research.

To be clear, the application of BPM in discrete manufacturing in smart environments is proposed as an addition, rather than as replacement of existing approaches and techniques, with the aim to enhance current practices where process-oriented approaches are desired. In the next section we outline the scope of this research, whereas the state-of-the art of systems and technologies in discrete smart manufacturing is discussed in Section 2.1.

1.5 Research scope

The proposition of applying BPM in smart manufacturing is broad and thus, scoping is required. The current section discusses the focus of this research in different aspects, more specifically the manufacturing sector, the functional hierarchy and the BPM lifecycle.

1.5.1 Manufacturing sector

The manufacturing sector is very diverse, combining activities with relatively low apparent labor productivity and average personnel costs, such as the manufacture of wearing apparel, wood products, furniture, and textiles, with other activities that have considerably higher values for the same indicators, such as manufacture of basic pharmaceutical products and pharmaceutical preparations, refined petroleum products and the manufacture of tobacco products. According to the statistical classification of economic activities in the European Community (NACE[12]), there are 24 different manufacturing subsectors, with the largest ones in terms of value added the manufacture of machinery and equipment and manufacture of motor vehicles and (semi-)trailers (data of 2018[13]). In European Union (EU), more than 2 million enterprises are classified as manufacturing in 2018, while the vast majority (1.96 million[14]) being small and medium-sized enterprises (SMEs). With more than 29.9 million people being employed in manufacturing in the EU in 2018, the manufacturing sector plays an important role in the economical and societal growth of many countries. The uptake of advanced technologies is growing but still in rather low levels – around only 30% of enterprises in the manufacturing sector in the EU27 zone have adopted advanced technologies[15]. While large manufacturers have the resources to invest in new technologies, the SMEs face difficulties to keep up with the technology trends and market competition (European Commission, 2021). This research targets SMEs, to support their efforts to integrate smart technologies in their operations. And while obtaining one AGV or a universal robotic arm can be affordable by a small manufacturer, there is often lack of knowledge and techniques how to smoothly integrate those in a set of operations (compared to a large enterprise which has, probably, already in place advanced systems to do so). Moreover, the focus of this research is the discrete production (as has already been mentioned in the previous sections), i.e., the manufacture of individual products or batches of individual products (countable pieces) (ORACLE, 2017), as SMEs in this domain face the most pressure from the mass customization and personalization trends.

1.5.2 Functional hierarchy

Regarding the operations and functions occurring in a manufacturing enterprise, there is a great range and therefore the current research needs to consider a relevant set of those. To do so, the widely adopted IEC 62264 (IEC, 2013b) international standard series is consulted. The series is a long-running development for pursue of integration of control systems to enterprise systems in the manufacturing domain. The first part (IEC 62264-1, also referred to as ANSI/ISA-95.00.01-2010) describes the interface content between manufacturing-control functions and other enterprise functions, based upon the Purdue reference model for computer integrated manufacturing (CIM) (Williams, 1990). The various types of control are classified in a functional hierarchy model, consisting of business planning and logistics, manufacturing operations and control, and batch, continuous or discrete control. The levels provide different functions and work in different time frames. At the bottom, Level 0 (not a control level) defines the actual physical process, i.e., the flow of material and products

[12] https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Glossary:Statistical_classification_of_economic_activities_in_the_European_Community_(NACE)
[13] https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Manufacturing_statistics_-_NACE_Rev._2
[14] https://www.statista.com/statistics/1252884/smes-in-europe-by-sector/
[15] https://ati.ec.europa.eu/data-dashboard/sectoral

throughout a factory. Level 1 defines the activities involved in sensing and manipulating the physical processes. This level typically operates on time frames of (milli-)seconds. Level 2 is concerned with the activities of monitoring and controlling the physical processes (through the sensors and manipulators of Level 1). Level 2 typically operates on time frames of hours, minutes and (sub-)seconds. Level 3 defines the workflow to produce the desired (end-) products. It includes activities of coordinating the processes and maintaining records. Level 3 typically operates on time frames of days, shifts, hours, minutes and seconds. On top, Level 4 defines the business-related activities needed to manage a manufacturing enterprise. Manufacturing-related activities include establishing the basic plant schedule, determining the inventory levels and making sure that materials are delivered on time to the right place. Level 4 typically operates on time frames of months, weeks and days. Activities of Levels 0, 1 and 2 are of less interest from a BPM perspective, as they represent the control of the physical aspects of manufacturing (i.e., the actual work performed by humans and machines). On the other hand, BPM is suitable for enterprise-level activities and is often employed to cover Level 4. Level 3 activities and information flows are defined by Manufacturing Operations Management (MOM) terminology and is the level of interest for this research, as the operations and processes on this level are the main concern for applying and extending BPM approaches.

The scope of the current research, with respect to the functional hierarchy of the IEC 62264-1 standard, is shown in Figure 7. As the end-to-end processes (e.g., including order reception and product delivery) are in concern, integration to Level 4 is also taken into account (only processes that are directly connected to production processes, e.g., order processing, are considered, excluding business-related ones such as account management, sales support or even product design). Accordingly, as the developed MPMS should be part of a CPS, integration to Level 2 is examined.

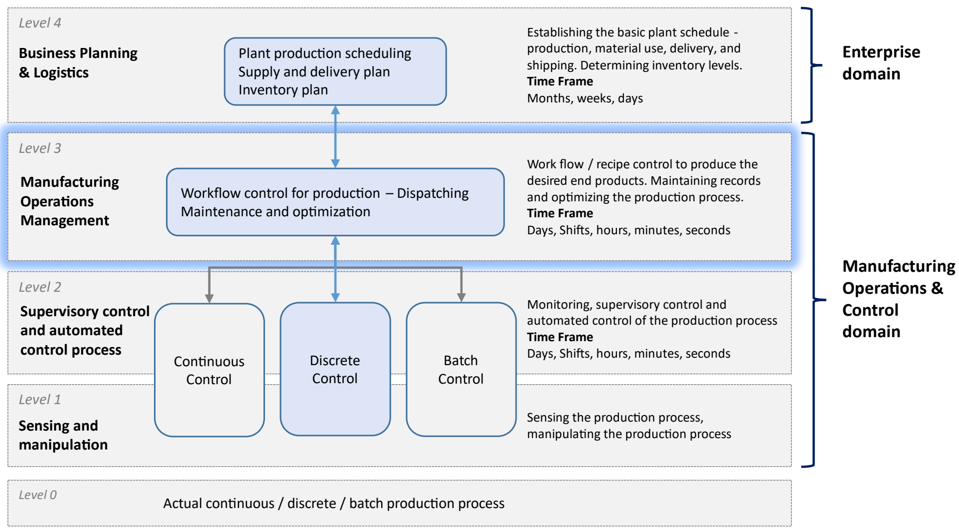


Figure 7: Functional domains and hierarchy of control in manufacturing (according to (IEC, 2013b)), with the main focus of the current research on activities and interfaces related to Level 3 in discrete production environments.

According to IEC 62264-1 standard, MOM shall be modelled using four categories: production operations management, quality operations management, inventory operations management, and maintenance operations management. This research is mainly concerned with the production operations management, which typically covers the largest part of operations, belongs merely to Level 3 and inherently has a process perspective.

1.5.3 BPM lifecycle

Engaging BPM in an enterprise helps to manage processes, actors and information. But which processes should be managed? In other words, which sequence of events and decisions are important to lead to a desired outcome? How far has a process been implemented? How is a process being controlled and monitored, so possible improvements can be identified? Such questions indicate that processes can be in different phases, typically with a cyclical link. Dumas et al., (2018) provide an overview of such a lifecycle, as shown in Figure 8. Adopting BPM starts with the process identification phase, when a business problem or need is posed. Processes related to the problem being addressed are identified. Those are then captured in as-is process models during the process discovery phase. In the subsequent process analysis phase, issues on the current processes are identified and documented, and whenever possible quantified with performance measures. Next, changes are identified in order to solve the issues and improve the as-is processes. Changes are then implemented to move processes to the desired, to-be state. Once the re-designed processes are running, relevant data are collected and analyzed to determine how well processes are performed. New issues, bottlenecks and deviations from the performance measures are identified and corrective actions are taken, which might mean that new or other affected processes have to be re-discovered.

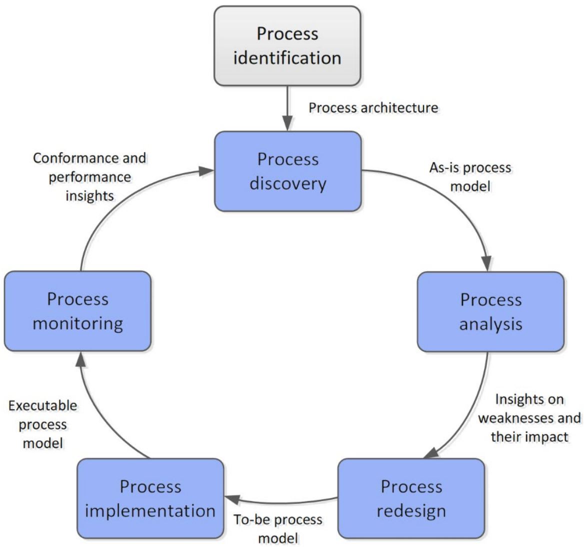


Figure 8: BPM lifecycle (according to (Dumas et al., 2018)) with the phases in scope highlighted in blue.

Assuming that manufacturers have already posed their problems and needs, and have identified what has to be done to address those, the process identification phase is not relevant and thus out of scope of the current research. All the rest of the phases of the above BPM lifecycle are of interest and this research intends to cover them. In case an organization has already started engaging BPM, there might already exist as-is process models in a notation that the current research considers, and thus the application of BPM can "start" from the process analysis phase. However, as there are not many BPM initiatives concerning manufacturing and production processes, most likely as-is process models have to be designed with the notation and methods explained in this research.

Lastly, apart from the three scoping aspects that are discussed above, it should be noted that this research does not put strong emphasis on aspects such as cyber security, system performance and robustness (the actual evaluated aspects are discussed in Section 7.2). The reason is that as the objective is to examine the application of an existing paradigm in new problems in an application domain that has not been (extensively) applied so far (as further discussed in Section 1.6), the emphasis is put on the functionality of the developed MPMS, rather than on its optimal utilization, which can be future work.

1.6 Research Objective

As already discussed, BPM is well-established and has proven its strength in various domains, but the adoption in manufacturing is not extensive, let alone at a mature level. Due to the nature of the paradigm and the domain (e.g., BPM focuses more on the information processing, ignoring in most cases the physical aspect, which is predominant in manufacturing), current BPM techniques and tooling are not well suitable for manufacturing. Adaptations and extensions are needed to support manufacturing operations. Moreover, the suggested solutions (Section 1.4) should consider the new problems that the current market trends and the advent of new technologies pose into the manufacturing domain (e.g., complex production scenarios or integration of autonomous robotic devices as introduced in Section 1.3). According to Gregor & Hevner (2013), this type of research, i.e., the adoption and extension of know solutions from another domain to solve new problems in a given domain, is referred to as *exaptation*. However, considering that new problems in the domain (e.g., increased collaboration between humans and robots) require new solutions (e.g., new modeling constructs), the research can be referred to as *invention*. Thus, this research is positioned, in the spectrum of the four types of design science research, at the intersection of invention and exaptation, as shown in Figure 9.

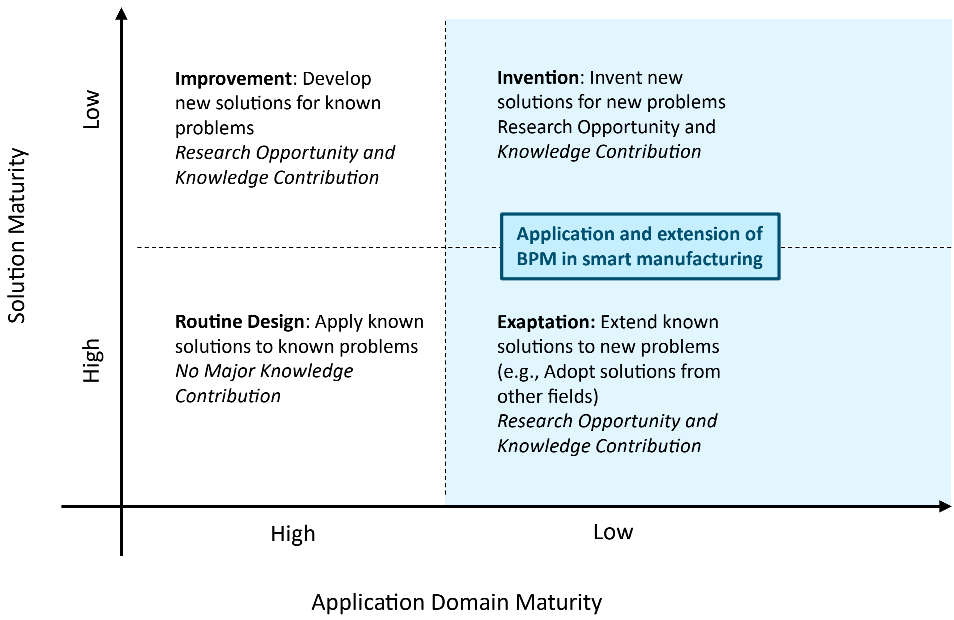


Figure 9: Positioning current research on the the design science research knowledge contribution framework (per (Gregor & Hevner, 2013)).

There has already been work on extending BPM for application in manufacturing. For instance, there are approaches to use BPMN for modeling manufacturing processes (e.g., (Zor et al., 2011; Prades et al., 2013; Abouzid & Saidi, 2019)), but they do not provide execution support. Erasmus' work (2019) covers execution aspects, but the work has to be complemented (as shown in Figure 6). Extensive discussion on existing work is provided in Section 2.1.7 and in Chapters 3, 4 and 5 where the individual developments of the current research are elaborated. Here, the main contributions that the current research adds are summarized:

- Modeling of manufacturing processes is approached from a wide perspective in order to cover as many manufacturing operations scenarios as possible.
- BPM is applied to a wide extent, covering many aspects of the paradigm (e.g., modeling, inclusion of resources/participants, exception handling, integration, technologies/systems to support the runtime execution).
- The theoretical concepts and tooling take into account that solutions should support integration to cyber-physical systems, to enable both horizontal and vertical integration.
- The theory and technology are applied to and evaluated with real-world manufacturing cases, within large European research and innovation projects.

The objective of this research, which leads to the above research significance, can be summarized in the following:

The objective of this research project is to provide models/constructs, guidelines and specifications of systems to apply advanced process management in smart manufacturing to tackle process complexity.

Research objective

1.7 Research design

To achieve the research objective, the research framework of Verschuren & Doorewaard (2010) is followed, which is a schematic and highly visualized representation of the phases/steps that need to be followed. Figure 10 illustrates the approach in 4 broad phases. In phase (a), an analysis phase, process modeling theory is confronted with real-world processes, as those being analyzed in practice, resulting in modeling constructs for representing complex processes. Similarly, both theoretical and practical analysis of exception handling in smart manufacturing is performed, resulting to a categorization of exceptions and handling strategies. For providing operational support, a manufacturing process management system (MPMS) is designed and implemented based on theory and standards on process integration. In phase (b), a design and implementation phase, the modeling constructs are applied to complex production scenarios and implemented in the system. Accordingly, the developed categorization for exception handling is mapped out on practical scenarios with the support of MPMS. The applied modeling constructs and the exception handling classification are evaluated in phase (c), i.e., an evaluation phase. The realized system, which incorporates all designs, is evaluated as well. Evaluation leads to the final artefacts of the given research objective.

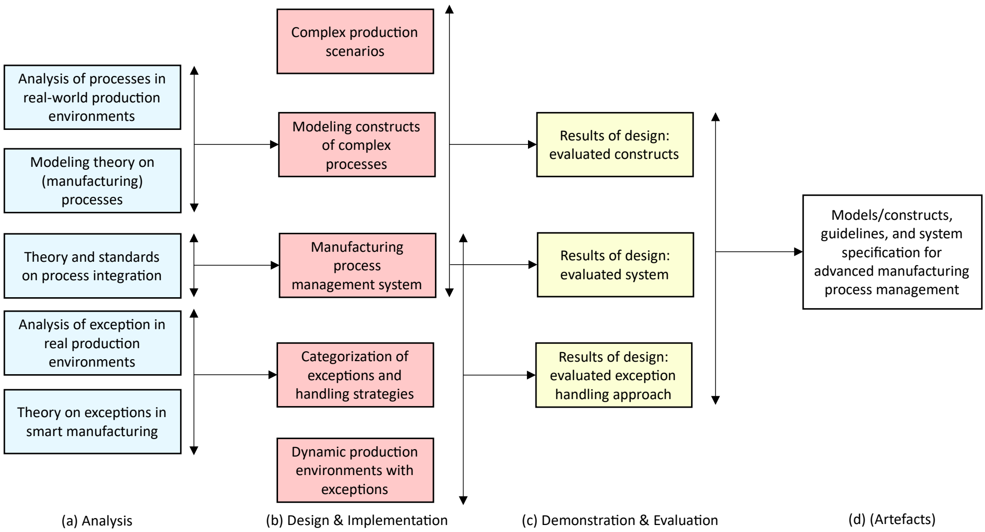


Figure 10: Research framework (based on (Verschuren & Doorewaard, 2010)) visualizing the steps to achieve the research objective.

The research framework helps to identify the research question(s) whose answers yield information that is necessary for accomplishing the research objective. The research objective, as formulated based on the problem statement, yields the following main research question:

Research Question

How can manufacturers tackle the process complexity in dynamic, discrete, smart production environments, in terms of flexible modeling and responsive enactment of their processes?

To derive an answer to this question, the following sub questions have been identified with the help of the above research framework (by subdividing the research framework into identifiable components):

RQ1: How can we provide flexible modeling of complex production processes?
RQ2: How can events and exceptions be handled in dynamic manufacturing environments?
RQ3: How can we enable process integration for end-to-end manufacturing process management?
RQ4: How can an advanced manufacturing process management system support the complexity tackling in smart manufacturing environments?

As this research aims to create and evaluate information systems artefacts with the ultimate purpose to solve practical needs within an organizational context, it follows the *design science* paradigm (March & Smith, 1995). As such, the activities to perform the research are guided based on the Design Science Research Methodology (DSRM) of Peffers et al. (2007) which consists of six main activities: *motivation and problem identification*, *solution objectives definition*, *design and development*, *evaluation*, and finally, *communication*. The mapping of the current research on DSRM is shown in Figure 11, with a rather straightforward sequence.

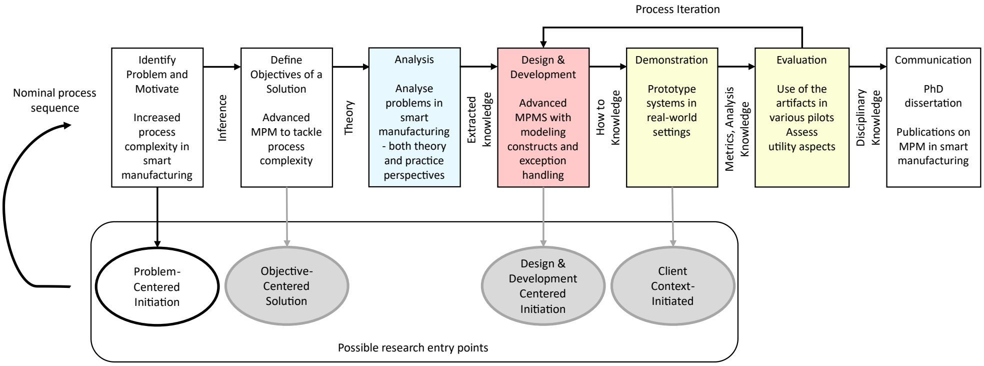


Figure 11: Design science research methodology (DSRM) process model (based on (Peffers et al., 2007)) as applied in the current research.

Regarding the second activity, the purpose of solving practical problems dictates that the objective of the solutions should be to generate knowledge that can be applied by users, in other words, to create utility (Gregor & Hevner, 2013). In the nominal process sequence, an *analysis* activity has been included after the definition of the objectives to extract the knowledge required for the actual design and development of the artifacts. Note that this analysis activity is not a new activity compared to the original DSRM (which is implicitly incorporated either in the problem identification or the design and development activities (Peffers et al., 2007)), but it is added to highlight what has to be analyzed. With respect to the research entry point, the current research follows a problem-centered approach. The colored phases correspond to the phases of Figure 10. Regarding the communication activity, apart from the current dissertation, a set of related scientific publications has been produced, listed in Table 1.

Table 1: Scientific publications related to this dissertation (listed per chronological order – newest first).

Chapter/Section	Article
Chapter 3	(Pantano et al., 2022) Pantano, M., Pavlovskyi, Y., Schulenburg, E., Traganos, K., Ahmadi, S., Regulin, D., Lee, D., Saenz, J., Pini, F., Francalanza, E., & Fraboni, F. (2022). **Novel Approach using Risk Analysis Component to Continuously Update Collaborative Robotics Applications in the Smart, Connected Factory Model**. Applied Sciences 2022, Vol. 12, Page 5639, 12(11), 5639. https://doi.org/10.3390/APP12115639
Chapter 8	(Grefen et al., 2022) Grefen, P., Vanderfeesten, I., Traganos, K., Domagala-Schmidt, Z., & Vleuten, J. van der. (2022). **Advancing Smart Manufacturing in Europe: Experiences from Two Decades of Research and Innovation Projects**. Machines 2022, Vol. 10, Page 45, 10(1), 45. https://doi.org/10.3390/MACHINES10010045
Chapter 5 / 6 / 7	(Traganos et al., 2021) Traganos, K., Grefen, P., Vanderfeesten, I., Erasmus, J., Boultadakis, G., & Bouklis, P. (2021). **The HORSE framework: A reference architecture for cyber-physical systems in hybrid smart manufacturing**. Journal of Manufacturing Systems, 61 (November 2020), 461–494. https://doi.org/10.1016/j.jmsy.2021.09.003
Chapter 3 / 6	(Traganos, Spijkers, et al., 2020) Traganos, K., Spijkers, D., Grefen, P., & Vanderfeesten, I. (2020). **Dynamic Process Synchronization Using BPMN 2.0 to Support Buffering and (Un)Bundling in Manufacturing**. *Lecture Notes in Business Information Processing*, *392 LNBIP*, 18–34. https://doi.org/10.1007/978-3-030-58638-6_2
Chapter 3 / 6 / 7	(Traganos, Vanderfeesten, et al., 2020) Traganos, K., Vanderfeesten, I., Grefen, P., Erasmus, J., Gerrits, T., & Verhofstad, W. (2020). **End-To-End Production Process Orchestration for Smart Printing Factories: An Application in Industry**. *Proceedings - 2020 IEEE 24th International Enterprise Distributed Object Computing Conference, EDOC 2020*, 155–164. https://doi.org/10.1109/EDOC49727.2020.00027
Chapter 7 / 8	(Erasmus, Vanderfeesten, Traganos, Keulen, et al., 2020) Erasmus, J., Vanderfeesten, I., Traganos, K., Keulen, R., & Grefen, P. (2020). **The HORSE Project: The Application of Business Process Management for Flexibility in Smart Manufacturing**. *Applied Sciences*, *10*(12), 4145. https://doi.org/10.3390/app10124145
Chapter 3	(Erasmus, Vanderfeesten, Traganos, & Grefen, 2020) Erasmus, J., Vanderfeesten, I., Traganos, K., & Grefen, P. (2020). **Using business process models for the specification of manufacturing operations**. *Computers in Industry*, *123*, 103297. https://doi.org/10.1016/J.COMPIND.2020.103297

Chapter 6 / 7	(Vanderfeesten et al., 2019) Vanderfeesten, I., Erasmus, J., Traganos, K., Bouklis, P., Garbi, A., Boultadakis, G., Dijkman, R., & Grefen, P. (2019). **Developing Process Execution Support for High-Tech Manufacturing Processes**. *Empirical Studies on the Development of Executable Business Processes*, 113–142. https://doi.org/10.1007/978-3-030-17666-2_6
Chapter 5	(Erasmus, Vanderfeesten, Traganos, & Grefen, 2018) Erasmus, J., Vanderfeesten, I., Traganos, K., & Grefen, P. (2018). **The case for unified process management in smart manufacturing**. *Proceedings - 2018 IEEE 22nd International Enterprise Distributed Object Computing Conference, EDOC 2018*, 218–227. https://doi.org/10.1109/EDOC.2018.00035
Chapter 6	(Erasmus, Vanderfeesten, Traganos, Jie-A-Looi, et al., 2018) Erasmus, J., Vanderfeesten, I., Traganos, K., Jie-A-Looi, X., Kleingeld, A., & Grefen, P. (2018). **A Method to Enable Ability-Based Human Resource Allocation in Business Process Management Systems**. *In: Buchmann R., Karagiannis D., Kirikova M. (eds) The Practice of Enterprise Modeling. PoEM 2018. Lecture Notes in Business Information Processing, vol 335. Springer, Cham*. https://doi.org/10.1007/978-3-030-02302-7_3
Chapter 5	(Erasmus, Grefen, et al., 2018) Erasmus, J., Grefen, P., Vanderfeesten, I., & Traganos, K. (2018). **Smart Hybrid Manufacturing Control Using Cloud Computing and the Internet-of-Things**. *Machines 2018, Vol. 6, Page 62*, *6*(4), 62. https://doi.org/10.3390/MACHINES6040062
Chapter 3	(Polderdijk et al., 2017) Polderdijk, M., Vanderfeesten, I., Erasmus, J., Traganos, K., Bosch, T., Rhijn, G. van, & Fahland, D. (2017). **A Visualization of Human Physical Risks in Manufacturing Processes Using BPMN**. *In: Teniente E., Weidlich M. (eds) Business Process Management Workshops. BPM 2017. Lecture Notes in Business Information Processing, vol 308. Springer, Cham*. https://doi.org/10.1007/978-3-319-74030-0_58

To ensure scientific rigor and practical relevance of our research, the IS design science research framework (DSR) from Hevner et al. (2004) is adopted. The framework is used to structure the core concepts of this research, the existing or extracted (after analysis) knowledge, the research activities (which follow the adapted DSRM), the developed artefacts and the produced knowledge. Figure 12 presents the symbols used in the representation of the framework.

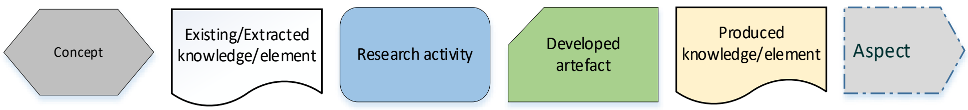


Figure 12: Legend for symbols used in the design science research framework of Figure 13.

The DSR structure for this research is shown in Figure 13. The *Environment* lane provides the practical input elements such as the industry needs, the current practices and the identified issues. On the other hand, the *Knowledge Base* lane provides established scientific foundations and methodologies (e.g., process modeling notations). Both lanes also serve as places to position the outcome of the research. More specifically, practical insights are gained from the evaluated demonstrations of the implemented solutions. In addition, technical documentation and the developed software can be used for further application and extensions. Accordingly, the research generates useful knowledge that is added in the base for knowledge establishment and for future research (in the form of this thesis and the publications of Table 1).

In the *IS Research* lane (which, as already stated, follows the adapted DSRM), the artefacts that provide answers to the research questions are generated. Adhering to the naming convention proposed by March & Smith (1995) for defining types of artefacts, this research produces:

1. A set of modeling constructs to represent (complex) manufacturing operations processes.
2. A categorization of exception types appearing is smart production environments and set of guidelines to determine suitable handling approaches.
3. A specification of an information system to design and enact manufacturing processes, as part of a CPS.
4. An architecture model of an advanced manufacturing process management system that integrates the first three design artefacts.

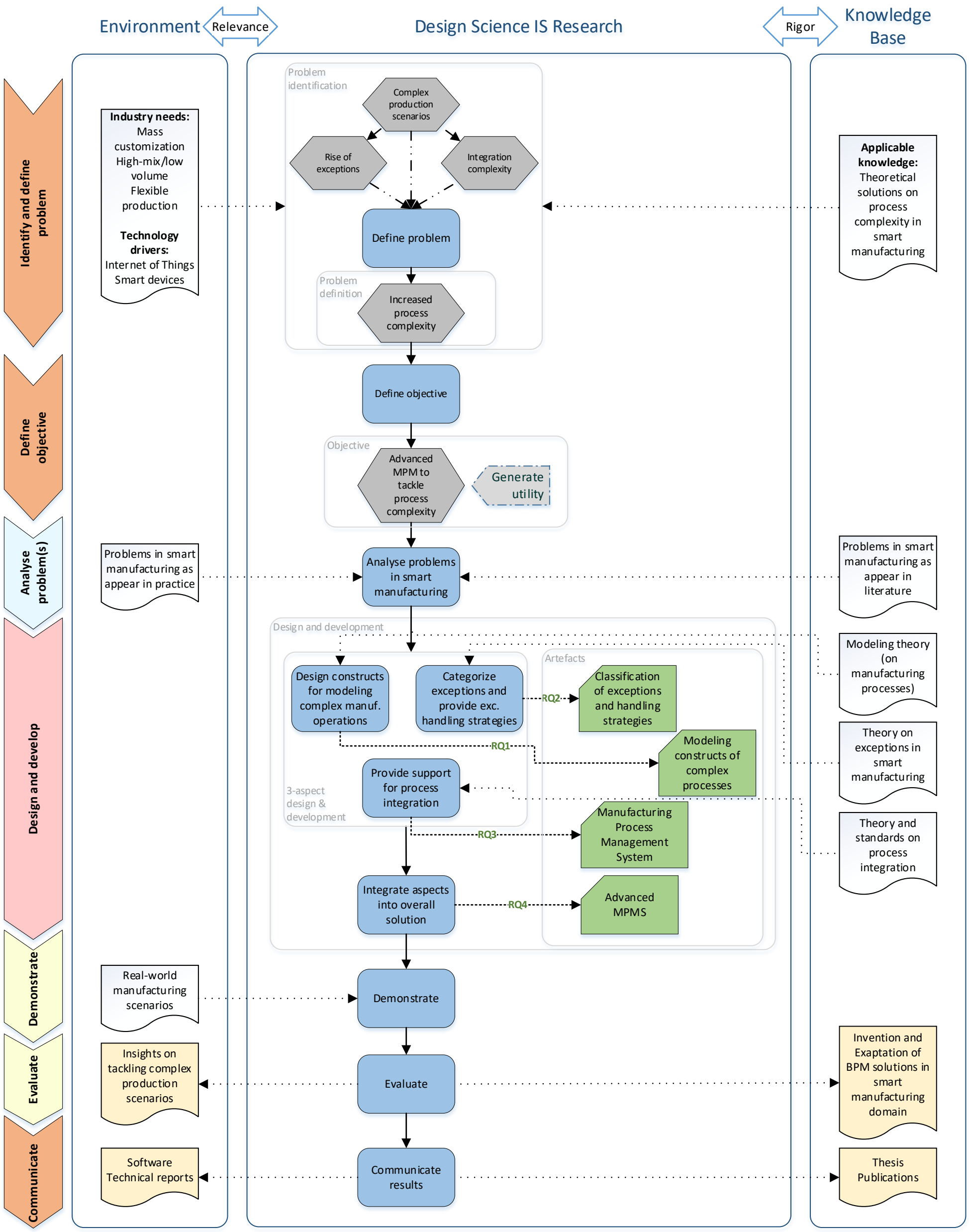


Figure 13: Information systems design science research framework (per (Hevner et al., 2004)), as applied in the current research, following the DSRM process model of Figure 11.

Regarding the significance of this research, the DSR knowledge contribution framework by Gregor & Hevner (2013) is referred. According to the framework, the knowledge produced through DSR can be either *descriptive* or *prescriptive*. As the outcomes of the current research are models, constructs and methods to support manufacturing operations, the contribution is considered as prescriptive. Concerning the level of knowledge contribution (all types shown in Table 2), all four artefacts are considered as *nascent design theory* (Level 2), as they generate knowledge as operational principles and architecture model. An instantiation outcome (Level 1) has also been generated for application in real-world settings and final evaluation of utility aspects. Together all four artefacts form a design theory[16] (Level 3 contribution) to provide enough knowledge on how to tackle process complexity in smart manufacturing with the BPM paradigm.

Table 2: Contribution types for design science research (Gregor & Hevner, 2013).

	Contribution types	**Example artefacts**
More abstract, complete, and mature knowledge	Level 3. Well-developed design theory about embedded phenomena	Design theories (mid-range and grand theories)
	Level 2. Nascent design theory—knowledge as operational principles/architecture	Constructs, methods, models, design principles, technological rules.
More specific, limited, and less mature knowledge	Level 1. Situated implementation of artefact	Instantiations (software products or implemented processes)

1.8 Thesis outline

The thesis chapters follow the same structure as the DSRM of Figure 11 (except the *Communication* activity which has resulted in the current dissertation, together with a list of related publications), shown in Figure 14.

The current chapter has introduced the context of this research, the problem motivation and the solution proposition. The design approach with the defined research objective, the identified research questions and the research methodology is presented as well.

Chapter 2 discusses the problem analysis from two perspectives. First, it provides a state-of-the-art overview of the status, theories and technologies seen in smart manufacturing. Second, it describes real-world problems encountered in manufacturing organizations. The identified problems appearing in practice are confronted with the available solutions to define the requirements of the proposed solutions.

Chapter 3 presents the details of the first designed artefact that deals with the process modeling of complex production scenarios. The developed BPMN 2.0 patterns and mechanisms are explained thoroughly, providing answers to RQ1.

Chapter 4 examines the exceptions that appear in smart manufacturing environments, with the goal to provide a classification for more effective identification. As each exception type typically requires a different handling strategy, a method to select one is elaborated. The second developed artefact answers RQ2.

[16] *Design theory*, as the fifth of the five types of theory in Gregor's (2006) taxonomy, gives prescriptions for *design and action*: it says how to do something.

Chapter 5 presents the architecture of MPMS, its main functionality and the integration specifications to other systems in the context of a complete CPS. The architectural model, as the third artefact of the current thesis, responds to RQ3.

Chapter 6 provides the consolidation of the first three artefacts into an architecture model of an advanced process management system (fourth artefact). It also presents a system instantiation, as developed to demonstrate feasibility and effectiveness of the models. The advanced MPMS is considered as the response to RQ4.

Chapter 7 presents the demonstration of the advanced MPMS in real-world use cases. It also discusses the evaluation of the system, as a verification of the solutions to solve the problems discussed in Chapter 2.

Chapter 8 concludes this thesis by, first, reflecting on the developed solutions against the identified problem(s) (as introduced in Chapter 1 and elaborated in Chapter 2), validating in that way the relevance of this research. After discussing limitations of the current research, the chapter, then, outlines future research. Finally, it summarizes key take-away messages.

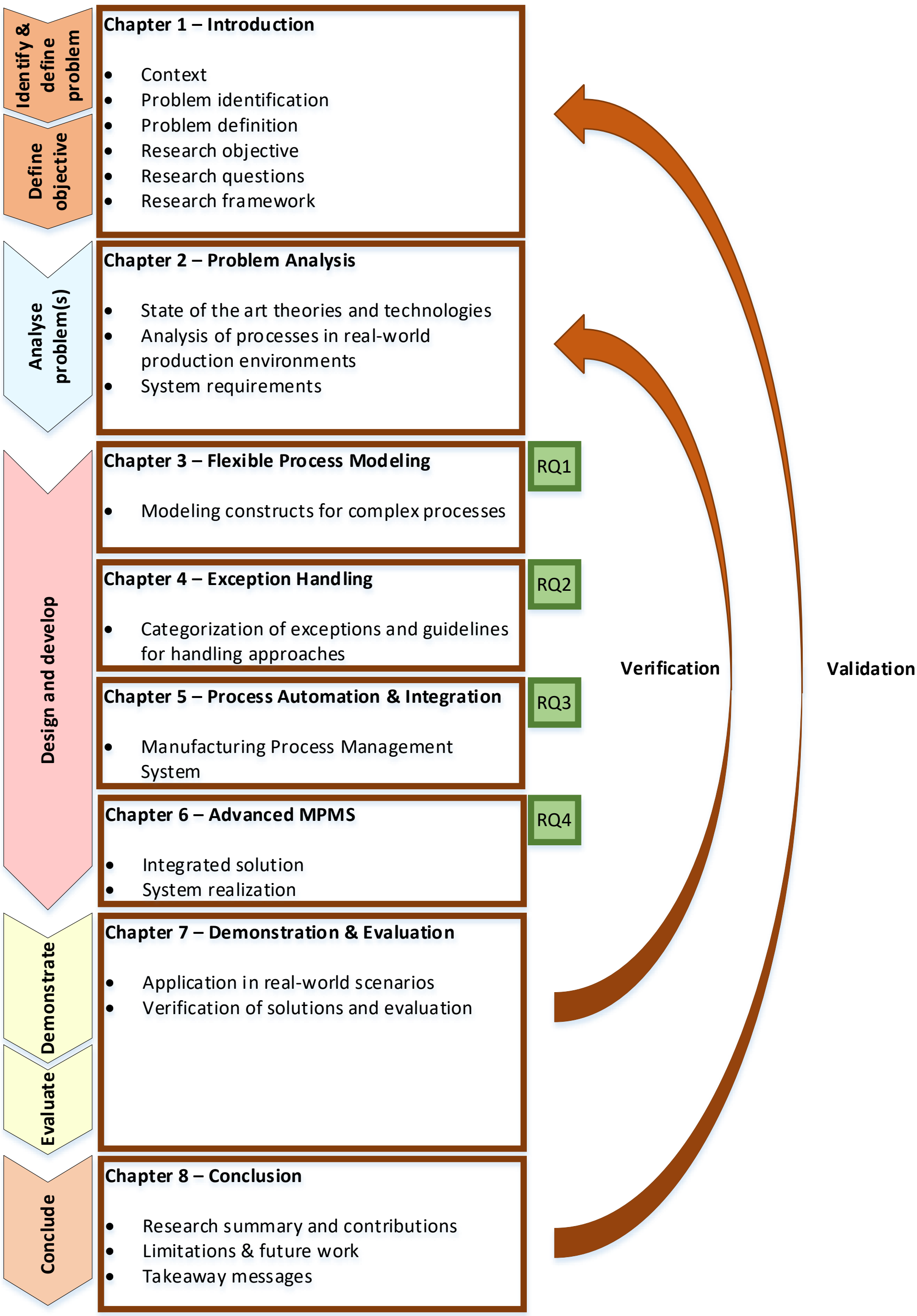


Figure 14: Schematic representation of the structure of the thesis.

CHAPTER 2

Problem Analysis

The current research focuses on solutions to tackle the operations complexity encountered in manufacturing environments, on the efforts of transforming traditional factories into smart ones from the process perspective. As it follows the DSR paradigm, as explained in Section 1.7, it must be grounded on existing scientific knowledge and be based on relevant problems appeared in practice. Therefore, this chapter presents the analysis of the problems to be solved from two perspectives: theory and literature on the one hand (Section 2.1), and practical situations from real-world use cases (Section 2.2). Both referring to the research context of smart manufacturing, from the scope of operations management (as delineated in Section 1.5). The problems and situations from literature and practice are then consolidated to form the requirements that the developed artefacts of this research should satisfy (Section 2.3).

2.1 State of manufacturing

This section provides an overview of the state of manufacturing, by providing relevant background information and discussing current approaches to dealing with process complexity. As understanding relevant concepts is important for the problem analysis and solutions design, the section first provides a classification of production, types of manufacturing systems, concepts of manufacturing operations management, and relevant information systems that support the operations. Then, the latest developments in the smart manufacturing era are presented. Current challenges and problems are examined, discussing also approaches to face them. With these problems identified from literature in mind, Section 2.2 presents them as identified in practical cases.

2.1.1 Typologies of production systems

A few decades ago, and especially towards the era of the third industrial revolution, various typologies of production and operations management systems had been developed with the intent to provide a meaningful methodology reappraisal, integration, and synthesis within production and operations management (Adam, 1983). Each of those have been based according to different dimensions and viewpoints. One of the first and highly influential typology of productions systems was provided by Joan Woodward (1965), who classified production according to technical complexity in ten classes, grouped in three main categories (from low to high technical complexity): *Small batch and unit production* (e.g., production of single pieces or fabrication of large equipment in stages), *large batch and mass production* (e.g., production of identical products in large numbers), and *continuous process production* (e.g., production of liquids and gases). Hull & Collins (1987) revised Woodward's typology by introducing the knowledge complexity (i.e., the technical expertise that is manifested in human knowledge and computers) as the criterion variable to subdivide the original batch category into *traditional* and *technical batch* (e.g., the production of an aircraft requires many highly trained workers). Hayes & Wheelwright (1984) used the process life cycle as the dimension to categorize production in *jumbled flow (job shop)*, *disconnected line flow (batch)*, *connected line flow (assembly line)* and *continuous flow* systems.

Kim & Lee (1993) included technical flexibility, i.e., the ability of a manufacturing system to cope with changing circumstances (Gupta & Goyal, 1989; Gupta & Somers, 1992), as a dimension to categorize production systems, which together with the technical complexity yielded a matrix of four possible production system types, shown in Figure 15. The four types are briefly explained below:

- *Intermittent production systems*: These systems retain technologies that are flexible in terms of production volume, product, expansion, machine, process, and routing, but do not have the capability of continuous use of facilities. The flow of the item being processed in such a production system is variable. Typical intermittent production systems are traditional job shops and batch processing systems (described briefly in Section 2.1.2).

- *Continuous production systems*: These systems retain technologies that are complex in terms of knowledge, automation, integration, and regulation, yet not flexible. The nature of the demand on such production systems that produce high-volume and standardized products results in continuous use of the facilities. Also, the material flow may be continuous as with automobile fabrication and assembly. The production process is integrated and makes use of mechanization and automation to achieve standardization and low cost. Typical systems in this class are assembly lines, transfer lines, and continuous flow processes.

- *Concurrent production systems*: These systems retain technologies that are complex in terms of knowledge, automation, integration, and regulation, yet have the capacity to produce small runs of different products. The advent of computer-aided manufacturing (CAM) systems, flexible assembly lines and FMS in general enabled the production of less standardized products, with concurrent activities and shorter manufacturing processes.

- *Degenerate production systems*: Systems that lack new process technologies, capital investment, intense supervision of labor, research and development activities, and the flexibility required to achieve low-cost production. A degenerate production system corresponds to a technologically inferior production system, non-competitive declining manufacturing system, and is characterized as “anachronistic factory”.

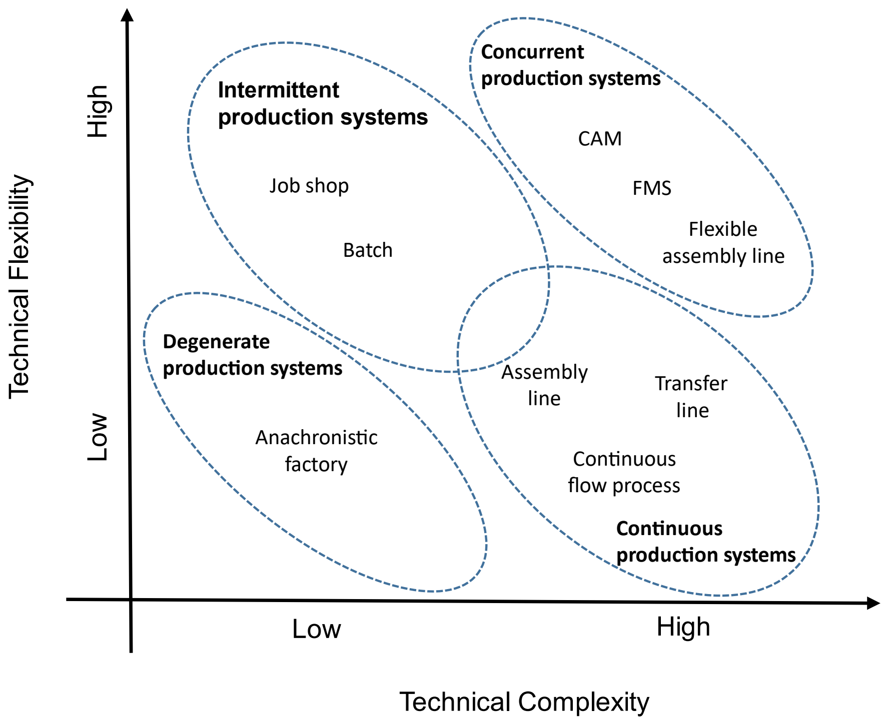


Figure 15: Production systems typology based on technical flexibility and technical complexity (per (Kim & Lee, 1993)).

According to IEC 61512-1 standard (IEC, 1997), also referred to as ANSI/ISA-88.01-1995 (ISA, 1995), industrial production processes can generally be classified as continuous, discrete parts manufacturing or batch:

- *Continuous processes*: In this type of processes materials are passed in a continuous flow through various processing equipment. Each piece of equipment typically performs one dedicated processing function. Once established in a steady operating state, the nature of the process is not dependent on the length of time of operations. The product output appears in a continuous flow and measured in amount/time. Oil refineries are typical examples for continuous production.

- *Discrete parts manufacturing processes*: In such processes, products are classified into production lots that are based on common raw materials, production requirements and production histories. In a discrete parts manufacturing process, a specified quantity of product moves as a unit (group of parts) between workstations, and each part maintains its unique identity. The product output is countable (in pieces). Consumer electronics (e.g., mobile phones) are typical products of discrete manufacturing processes.

- *Batch processes*: This type of processes leads to the production of finite quantities of material (batches) by subjecting quantities of input materials to a defined order of processing actions using one or more pieces of equipment. The product produced by a batch process is called a batch. Batch processes are neither discrete nor continuous; however, they have characteristics of both. The product output is measurable (e.g., in kilograms or liters), but it is typically difficult to maintain batch identity if common storage is used.

The word *production* has, in general, a broader meaning and covers all types of processes, compared to *manufacturing*, which typically refers to discrete or batch processes (for instance, the term “crude oil manufacturing” in a refinery plant, referring to a continuous process, does not make much sense). Thus, as this research is mostly focused on discrete parts or batch processes (as already mentioned in Section 1.5.2), the term manufacturing is heavily used. Technologically speaking, the term manufacturing, as defined by Groover (2010), refers to the application of physical and chemical processes to alter the geometry, properties, and/or appearance of a given starting material to make parts or products; it also includes assembly of multiple parts to make products. Manufacturing processes involve a combination of machinery, tools, power, and labor, with the intention to add value onto starting material. While the principle of transforming material in order to get (economic) value out of it is still valid in the modern types of manufacturing, it is the combination of all involved parts that is getting more complex and requires attention.

2.1.2 Manufacturing systems

The physical way manufacturing companies arrange their factory facilities and equipment is called *plant layout*, while the way they organize them into logical groupings is called *manufacturing systems*. Over the years, certain types of manufacturing systems have been well-established as the most appropriate way to organize production for a given combination of product variety and production quantity (Groover, 2010).

A *job shop* is a type of production facility that makes specialized and customized products in the low-quantity range (1-100 units/year), e.g., ships, aircrafts or special machinery. When the product is heavy and thus, hard to move, it typically stays in a single location during its fabrication and assembly. Such a *fixed-position layout* is shown in Figure 16(a) . In practice, the individual components are built at single locations in factories and brought together for the final assembly. In case the facilities and equipment are arranged according to type or function, the arrangement is called a *process layout*. An example is illustrated in Figure 16(b), where parts that require a different processing or operation sequence are routed through different departments in a particular order. It should be noted that the *process layout* should not be confused with the process perspective that this research mainly considers. Any type of plant layout involves processes. The product variety determines also the type of facilities in the medium-quantity range (100 – 10,000 units/year). To deal with a wide product variation, *batch production* is usually followed, in which the equipment is changed over between the production of batches of products. This type of production is commonly used for make-to-stock product delivery strategies (Olhager, 2003), in which items are manufactured and stored as intermediate or finished products. In case the product variation is limited, manufacturing systems are often configured as *cells*, consisting of several workstations and machines, where each cell is specialized in the processing or assembly of a given set of similar parts or products. The *cellural layout* is depicted in Figure 16(c) . Process and cellural layouts are typical layouts in mass production as well (i.e., 10,000 to millions of units/year). For products that their processing requires units and parts that are physically moved through a sequence of equipment and workstations, *product layouts*, as shown in Figure 16(d) , are the commonly arranged manufacturing systems. Car *assembly lines* are familiar examples of a series of connected line of segments.

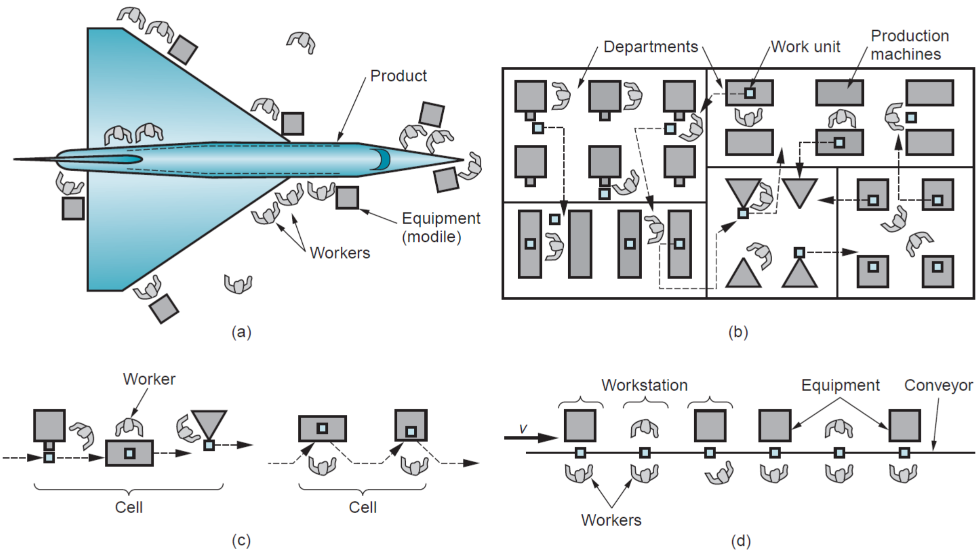


Figure 16: Various types of plant layout: (a) fixed-position layout, (b) process layout, (c) cellural layout, and (d) product layout (Groover, 2010).

Apart from the product variety and production quantity, other aspects play a determining role on the type of manufacturing systems. Low-cost production, enhanced product quality and rapid responsiveness to changes are main goals of every manufacturing enterprise and in turn its manufacturing systems (Koren, 2006). Cost-effective systems are the dedicated machining systems (DMS) that produce one specific part type at high volumes and the required quality (Mehrabi et al., 2000). They use transfer line technology (also referred to as dedicated manufacturing lines – DML) with fixed automation and tooling, i.e., product-specific machine tools (PSMT). DMS are driven by the economy of scale and are suitable for mass production, but they do not respond to market needs for smaller quantities of differing products. For such production cases, flexible manufacturing systems (FMS) have been introduced. These systems consist of computer numerically controlled (CNC) machines, and other fixed but programmable software to produce a variety of products on the same system. FMS provide a general flexibility through the use of equipment with built-in high functionality and shortened changeover times (el Maraghy, 2006). However, the equipment is typically expensive, and the production-rate is very small due to the single-tool operation. The category of systems that provide some flexibility at affordable costs and with acceptable productivity is the reconfigurable manufacturing systems (RMS). As Abele et al. (2006) classifies them (Figure 17), RMS lay between DMS and FMS with regards to productivity and flexibility aspects. They are designed to quickly adjust production capacity and functionality, within a part family (i.e., one or more part types with similar characteristics), in response to changes in market demands (Koren, 2006). They consist of reconfigurable machine tools (RMT) (Landers et al., 2001) which on a system level are linked into sequential or parallel production lines. Through the ability to add, rearrange, replace and remove components, RMS provide modularity and versatility of the machines.

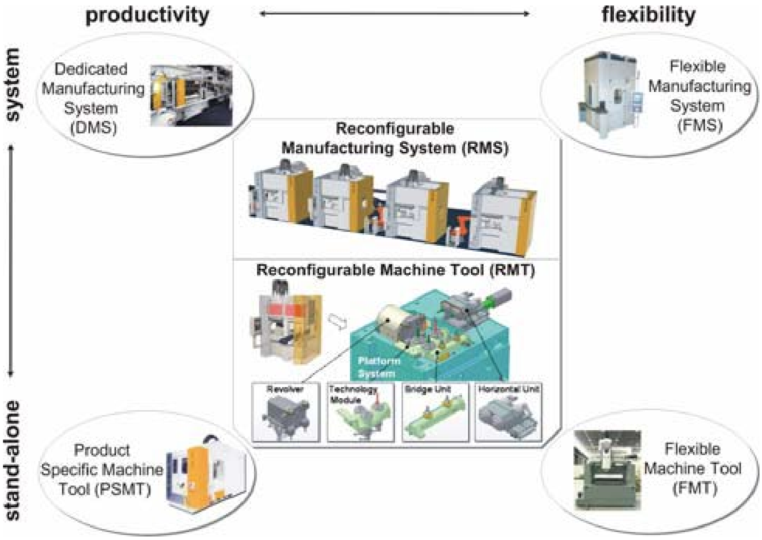


Figure 17: Classification of traditional manufacturing systems (Abele et al., 2006).

Regardless the plant layout and the characteristics of production, the equipment and assets of a manufacturing enterprise, which form its manufacturing systems, are usually organized in a hierarchical way, where lower-level groupings are combined to form higher levels. The IEC 62264-1 standard (IEC, 2013b) provides a reference model to describe such equipment hierarchies with a common terminology. An *enterprise* is a collection of *sites*, which in turn is a collection of *areas*. It is responsible to determine what products will be manufactured, at which sites and in general how they will be manufactured. A *site* is a physical, geographical, or logical grouping determined by the enterprise. An *area* is a physical, geographical, or logical grouping within a site and may contain *work centers*. In turn, *work centers* contain *work units*. That hierarchy is a role-based hierarchy, as the equipment model is defined in terms of performed functions and activities (described in Section 2.1.3) that equipment entities may perform. A physical hierarchy model can be then designed, including the specific assets. Depending on the type of production (i.e., batch, continuous, discrete), the terms *work centers* and *work units* get more specific. These types are shown in Figure 18. Thus, a *work center* can be a *process cell*, a *production unit*, or a *production line*. A *storage zone* is included to describe equipment for storage or movement. A *work unit* can be a *unit*, a *work cell* or a *storage unit*.

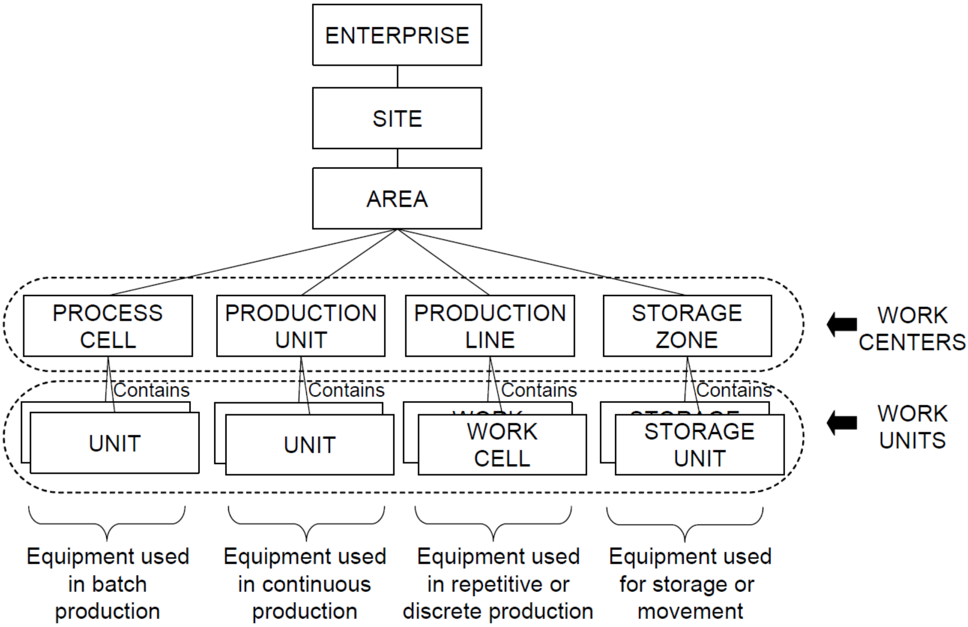


Figure 18: Types of work centers and work units according to the role-based equipment hierarchy of IEC62264-1 standards (IEC, 2013b).

2.1.3 Manufacturing operations management

Every manufacturing firm performs various production activities, together with enterprise ones that any business entity performs. The integration and collaboration of these two broad categories of functions, while crucial and beneficial, is often a challenge (Hausman et al., 2002; O'Leary-Kelly & Flores, 2002; Tang, 2010b). A very important first step is to identify and describe the boundaries between the enterprise domain and the manufacturing operations and control domain (Chen †, 2005). The IEC 62264 standard (IEC, 2013b), as already briefly introduced in Section 1.5.2, provides reference models to define and categorize functions and activities across those two domains, and to specify the information and data flow between corresponding systems.

The standard provides a functional hierarchy to classify the various types of control in manufacturing, as is shown in Figure 7. Manufacturing operations management (MOM), labeled as Level 3, is concerned with activities of a facility (Area level and levels below per the role-based hierarchy of Figure 18) that coordinates the personnel, equipment and material in manufacturing. These involve both physical activities and digital activities performed by information systems. Four categories of manufacturing operations are defined, each of which consists of main functions:

- Production operations
    - Production scheduling
    - Production control
- Quality operations
    - Quality assurance
- Inventory operations
    - Product inventory control

    - Material and energy control
- Maintenance operations
    - Maintenance

Of course, every manufacturing enterprise performs other supporting functions within manufacturing operations management, such as management of security or management of documents, etc., which are not the main interest of this research.

Within production operations management, production control, as the main focus of the current research, includes functions associated with manufacturing operations and control. These typically are: controlling the manufacturing of products from raw materials according to a production schedule, designs and standards, performing plant engineering activities, generating performance reports, evaluating capacity constraints. Production control also encompasses functions of process support engineering such as issuing requests for modification or maintenance, coordinating maintenance and engineering functions, and providing technical support to operators. It also includes operations planning functions such as setting-up a short-term production plan according to a production schedule, checking the schedule against equipment and personnel availability.

Regarding production scheduling, it is the level of detailing and time frame that make it part of productions operations management. Determination of detailed production schedule (i.e., which resource shall handle which productions activities) in short-term (e.g., days, shifts, hours, minutes), and refers to specific site and area is a Level 3 function. Broader scheduling on orders level, which happens on monthly or weekly basis and on enterprise or site level, is a Level 4 (enterprise level) function. Therefore, production scheduling functions are considered as interfacing functions between enterprise and manufacturing operations. Similarly, product inventory and material and energy control functions span over both enterprise and manufacturing and control levels. Other enterprise functions are order processing, procurement, product cost accounting, product shipping administration, market and sales, research and development. The set of main functions for both Level 4 (enterprise) and Level 3 (MOM) is illustrated (based on the Yourdon model notation (DeMarco, 1979)) in Figure 19. Data flows among these functions are also shown (omitting, at this point, discussion on what these data flows represent). It should be noted that the categorization and representation of functions do not reflect any organization structure. In other words, an enterprise might structure their organizational activities and departments in a different way than the operations clustering.

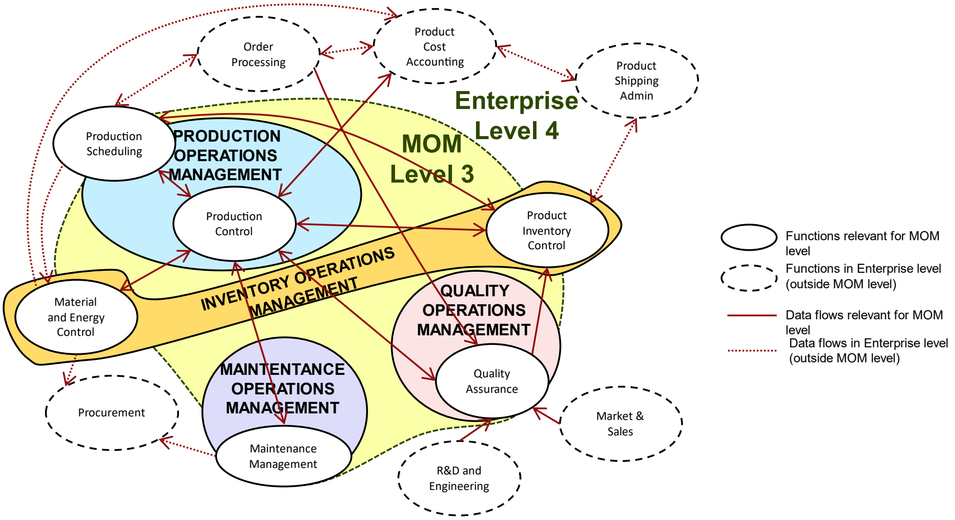


Figure 19: Simplified enterprise and control functions model (per IEC 62264-1 standard (IEC, 2013b)).

For each of the four main categories of MOM, an activity model is defined to further elaborate the operations and their functions, and the data flow between them. A generic model consists of the following activities (as collections of tasks): detailed scheduling, dispatching, execution, resource management, definition management, tracking, data collection and (performance) analysis. Specifying this for production operations management, the activity model of Figure 20 is developed. The arrowheads represent information flow, both between activities within Level 3 (MOM level) and between Level 3 and other levels. The IEC 62264-1 standard distinguishes four main categories of information exchanged between Level 4 and Level 3, namely definition information (i.e., what it takes to manufacture a product), capability information (i.e., what resources are available), schedule information (i.e., what to produce and use, and when to do so) and performance information (i.e., what was made and used). These categories, specified for production operations, are also shown in Figure 20.

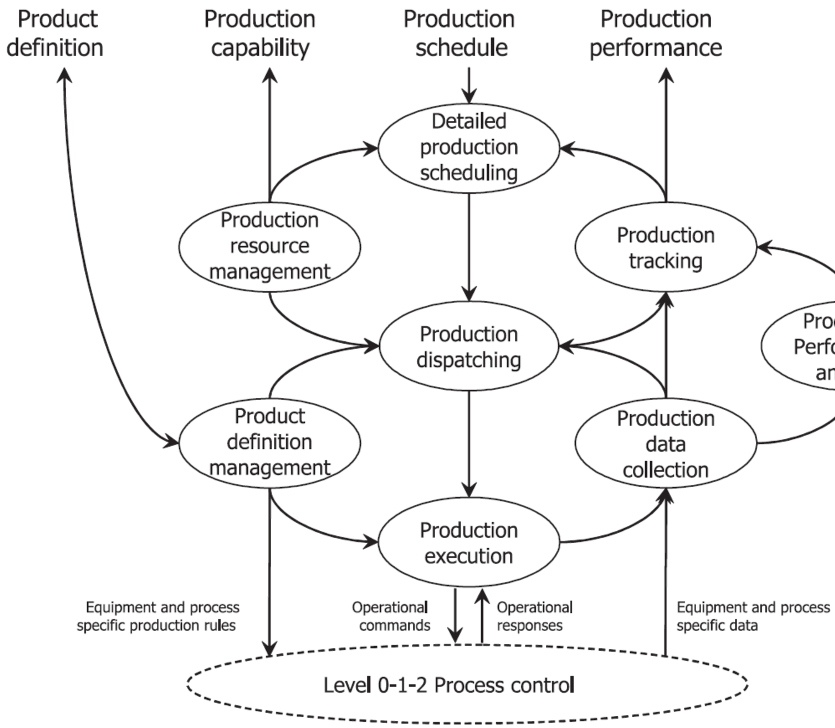


Figure 20: Activity model of productions operations management (per IEC 62264-1 standard, (Chen †, 2005)).

2.1.4 Manufacturing information systems

Various types of information systems have emerged to support the different functions and activities of manufacturing enterprises. Romero & Vernadat (2016) distinguish six main types of enterprise information systems (EIS): enterprise resource planning (ERP) systems, supply chain management (SCM) systems, manufacturing execution systems (MES), customer relationship management (CRM) systems, product lifecycle management (PLM) systems and business intelligence (BI) systems. Except MES, the rest types support business functions (enterprise domain) such as order processing, marketing and sales, product specifications data management, accounting and finance, etc. Other systems encountered for business functions support are the business process management systems (BPMS), which are considered as more mature Workflow Management Systems (WfMS). Each of these types specializes in particular functional areas, but all are considered to cover the Level 4 of the IEC 62264 standard (to be clear, the standard does not explicitly define the functions of Level 4 but rather assumes all business functions, which are not covered by the other levels, to be in that level).

In the control domain, the four main categories of operations (Figure 19), namely production, quality, inventory and maintenance, are primary supported by manufacturing execution systems (MES), quality management systems (QMS), warehouse management systems (WMS) and computerized maintenance management systems (CMMS). These systems cover Level 3 of IEC 62264 standard. On Level 2 of the standard, the most common control and automation systems that handle the hardware of Level 1 (e.g., actuators, sensors, input/output (I/O) devices) are: supervisory control and data acquisition (SCADA) system, programmable logical controller (PLC), computer numerical controller (CNC), distributed control system (DCS), batch automation system (BAS) and robot controller (Alexakos et al., 2006; Mehta & Reddy, 2015; Nagorny et al., 2012).

Figure 21 places the typical information systems in manufacturing on the functional levels of the IEC 62264 standard, as discussed above. This serves rather as an overview and not as a robust classification, especially when the advancements in EIS make them cover more and more functions. For instance, ERP systems have significantly expanded their scope over the last decades (Kurbel, 2013; Nwankpa, 2015; Rerup Schlichter & Kraemmergaard, 2010; Seethamraju, 2015). Rashid et al. (2002) consider them as software systems that integrate business processes including planning, marketing, sales, accounting, human resource management, e-business etc. Similarly, Monk & Wagner (2013) argue that ERP may support many functional areas. Accordingly, advanced MES from large vendors might integrate functionality of all types of operations. Moreover, as the boundaries between levels are not distinct, there can be systems with cross-level functionality, such as the manufacturing intelligence system proposed by Unver (2013) that contextualizes low-level shopfloor data using production operation information from ERP systems. It should be noted also, that typical BPMS cover Level 4 functions, while the current research investigates the exaptation of a BPM system to cover Level 3 functions.

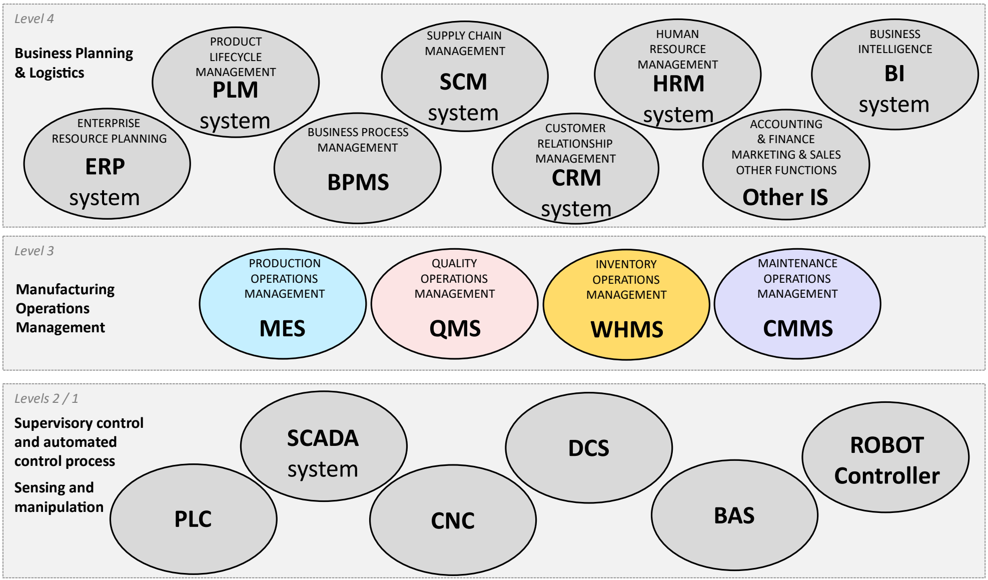


Figure 21: Typical information systems per functional control level and function areas.

MES, as the most dominant EIS for MOM, is worth further investigation. The Manufacturing Enterprise Solutions Association (MESA[17]) defines MES as a system that "*delivers information that enables the optimization of production activities from order launch to finished goods. Using current and accurate data, an MES guides, initiates, responds to and*

[17] https://mesa.org/

reports on plant activities as they occur" (MESA, 1997). MESA, through a survey on major actors of the market, gathered the following 11 functions of an MES (de Ugarte et al., 2009):

1. Operations/Detail Scheduling: sequencing and timing activities for optimized plant performance based on finite capacities of the resources.
2. Process Management: directing the flow of work in the plant based on planned and actual production activities. It should not be confused with the general business process management approaches that this research applies, as here process management covers only sequence of activities.
3. Document Control: managing and distributing information on products, processes, designs or orders, as well as gathering certification statements of work and conditions.
5. Data Collection/Acquisition: monitoring, gathering and organizing data about the processes, materials and operations from people, machines or controls.
6. Labor Management: tracking and directing the use of personnel during a shift based on qualifications, work patterns and business needs.
7. Quality Management: recording, tracking and analyzing product and process characteristics against engineering ideals.
8. Dispatching Production Units: giving the command to send materials or orders to certain parts of the plant to begin a process or step.
9. Maintenance Management: planning and executing appropriate activities to keep equipment and other capital assets in the plant performing to goal.
10. Product Tracking and Genealogy: monitoring the progress of units, batches or lots of output to create a full history of the product.
11. Performance Analysis: comparing measured results in the plant with goals and metrics set by the corporation, customers or regulatory bodies.
12. Resource Allocation and Status: guiding what people, machines, tools and materials should do, and tracking what they are currently doing or have just done.

The 11 functions are shown in Figure 22, where MES is presented as a full MOM system integrating shop floor data and ERP information, covering all four types of operations. Of course, not every MES system (has to) cover(s) all 11 functions and depending on the needs, extra focus is given on specific ones. Moreover, while the functions are still relevant in smart manufacturing, MES (whose origin dates back to mid-1990s (de Ugarte et al., 2009) or even in early 1980s (Kletti, 2007)) has to be adapted to Industry 4.0 concepts (Mantravadi & Møller, 2019). For instance, Kannan et al. (2017) propose, through model-based requirement modeling, a set of requirements for building an MES in automotive sector, compliant with current industry standards. Though, an MES implementation according to those requirements is not provided (and thus validating the requirements).

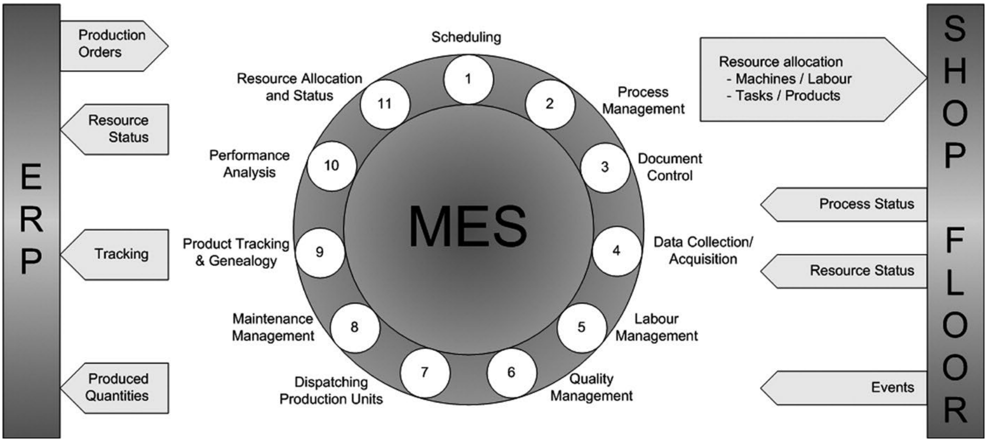


Figure 22: MES functionalities (de Ugarte et al., 2009).

2.1.5 Smart manufacturing

While the core concepts of manufacturing are still valid and applicable, the domain is going through disruptive changes on various aspects, resulting in many of the current systems to be obsolete or needing evolvement. On business aspects, there is a shift from traditional supply chain models that focus on the efficiency on the production side to demand chain models that put emphasis on the customer side. Manufacturers try to satisfy customers by providing highly customized or mass-personalized products and offering shorter delivery times, which requires of course operations flexibility (Thoben et al., 2017; Wang et al., 2017). Moreover, in their efforts to decrease (or even avoid) intermediate stock and react on last-minute changes due to dynamic market conditions, just-in-time (JIT) production models (Brox et al., 2010) become more important (Hofmann & Rüsch, 2017).

On technology aspects, manufacturing domain encounters growing developments and application of digital technologies (Lu, 2017), such as cloud computing (Shawish & Salama, 2014), internet-of-things (IoT) (Atzori et al., 2010; Lu & Cecil, 2015), artificial intelligence, big data. Devices and machines are equipped with a plethora of sensors that turn them into intelligent, context-aware and even self-controlled nodes of production systems, often placed in network setups. The coupling of digital systems to physical ones into cyber physical systems, shifts automated manufacturing towards intelligent manufacturing (Thoben et al., 2017). Moreover, advancements in robotics, such as collaborative robots (el Zaatari et al., 2019) and augmented reality (Nee et al., 2012) change the landscape of production systems for more agile manufacturing. The existing DMS are decaying in industries where highly customization is needed (Koren, 2006) and is expected that customized products will be manufactured by smart robotics acting in dynamic processes managed on cloud platforms (Zhang et al., 2014). On the other hand, FMS have not been widely adopted and many of the manufacturers that bought FMSs are not pleased with their performance (Koren, 2010). Manufacturers have to reorganize their production facilities to include more modern types of equipment and machinery.

The term smart manufacturing is used to characterize the current, new traits of manufacturing in the ongoing fourth industrial revolution, both from business and technology perspectives as briefly described above. Other terms widely encountered in literature to describe the technological progress of manufacturing are intelligent manufacturing, IoT-enabled manufacturing and cloud manufacturing. According to Zhong et al. (2017), intelligent manufacturing is regarded as "a *new manufacturing model based on intelligent science and technology that greatly upgrades the design, production, management, and integration of the whole life cycle of a typical product*". Thoben et al. (2017) stress the ability of manufacturing systems to "*self-regulate and/or self-control to manufacture the product within the design specifications*". The main technology in intelligent manufacturing is artificial intelligence, which enables production systems to adapt to market circumstances (Lu et al., 2016). The term typically focuses on technological aspects and not on organizational ones (Thoben et al., 2017). Thus, intelligent manufacturing is considered as a part of the broader smart manufacturing term, as has already been shown in Figure 3. The terms IoT enabled manufacturing and cloud manufacturing are heavily influenced by the underlying technologies. IoT-enabled manufacturing is based on the principle of converting production resources (devices, material and products) into smart manufacturing objects that have the ability to sense, connect and interact with each other to execute production activities (Zhong et al., 2017). In cloud manufacturing, cloud computing technology is applied. Wu et al. (2013) define it as "*a customer-centric manufacturing model that exploits on-demand access to a shared collection of diversified and distributed manufacturing resources to form temporary, reconfigurable production lines which enhance efficiency, reduce product lifecycle costs, and allow for optimal resource loading in response to variable- demand customer generated tasking*".

As the term *smart manufacturing* has a broader view, it is the one adopted in this research. Nevertheless, regardless the term, the modern type of manufacturing possesses characteristics and adheres to principles of the general Industry 4.0 developments, which are discussed below. As the current research is motivated by how manufacturers can overcome issues and adapt their traditional factories to embrace Industry 4.0 technologies, the concept of smart factory is analyzed as well.

2.1.5.1 Industry 4.0

Driven by market pull requirements, such as product individualization on demand, flexibility in product development, demand fluctuations, stricter regulations, and pushed by technology developments, such as smart devices, versatile and collaborative robots, increasing digitization, cloud computing (Ahuett-Garza & Kurfess, 2018; Lasi et al., 2014; Monostori, 2014), Industry 4.0 promises increased productivity, higher resources efficiency, flexibility and labor cost reduction (Dalenogare et al., 2018; McKinsey Digital, 2015; Pereira & Romero, 2017). Core components that realize such promises and goals are (Oztemel & Gursev, 2020): cyber-physical systems, cloud systems, machine to machine (M2M) communication, smart factories, augmented reality and simulation, (big) data mining, internet of things, ERP and BI systems, and virtual manufacturing technologies. Other fundamental features and concepts surrounding Industry 4.0 are self-organization/decentralization, modularity, interoperability, real-time capability, corporate social responsibility and sustainability (Carvalho et al., 2018; Lasi et al., 2014).

The term Industry 4.0, as one of the most used to describe the developments in the fourth industrial revolution, especially in Europe, collectively refers to a wide range of concepts. Lichtblau et al., (2015) nicely describe it as the "*fusion of the physical and virtual worlds*" – with digitization as the merging mechanism of the smart factory and smart products from the physical world with the smart operations and data-driven services from the virtual world – but an holistic and unanimous definition is rather hard to give. Though, several frameworks already exist to give structure in the broad Industry 4.0 term. A generally acceptable in the manufacturing domain framework is the Reference Architectural Model for Industry 4.0 (RAMI 4.0), established in 2015 (DIN/DKE, 2016; Hankel & Rexroth, 2015). The reference architecture describes all crucial elements of Industry 4.0 in a three-dimensional layer model, to break down complex interrelations and classify relevant technologies. More specifically, it relates *layers*, *life cycle & value stream* and *hierarchy levels*, illustrated by the three-dimensional cube shown in Figure 23 and briefly described below.

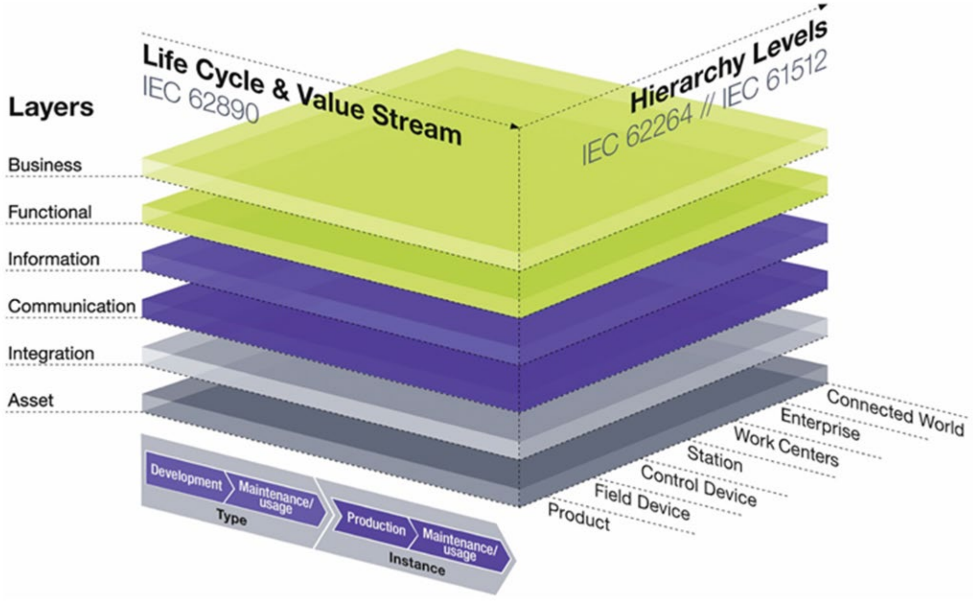


Figure 23: Reference Architectural Model Industry (RAMI) 4.0 (Hankel & Rexroth, 2015).

- The *layers* dimension represents the information that is relevant to the role of an asset. It covers the business-to-technology spectrum by relating different aspects of a manufacturing asset to layers of the enterprise architecture.

- The *life cycle & value stream* dimension represents the lifetime of an asset and the value-added process. This axis distinguishes between the type and instance of a production system and its elements, for example the digital design of a product and its (multiple) instantiation as a manufactured product.
- The *hierarchy levels* dimension is used to assign functional models to specific levels of an enterprise. This axis uses aggregation to establish enterprise levels, ranging from the connected world (i.e., networks of manufacturing organizations in their eco-systems) via stations (manufacturing work cells) to devices and products. The hierarchy levels dimension is related to the IEC62264-1 standard. The connected

world level is introduced above the enterprise level of the standard to emphasize the importance of supply chain networks in Industry 4.0. Additionally, lower levels are added to elaborate the control systems and equipment typically encountered in modern factories.

A few other approaches structure the concepts of Industry 4.0. The Industrial Internet Reference Architecture (IIRA), designed by the Industrial Internet Consortium, consists of four viewpoints to support the design and implementation of an Industrial IoT (IIoT) system (Lin et al., 2017). The Smart Manufacturing Ecosystem (Lu et al., 2016), developed by the National Institute of Standards and Technology (NIST), provides a complete overview of various lifecycles of smart manufacturing. The framework's standardization mainly focuses on ICT application systems and how information is exchanged among software applications, omitting infrastructures concepts like IoT and cloud computing (Li et al., 2018). Fraile et al. (2019) present and realize the Industrial Internet Integrated Reference Model (I3RM), which integrates features of NIST smart manufacturing standards, IIRA, and RAMI 4.0 reference models and architectural patterns, to facilitate the definition of the system architecture of digital manufacturing platforms. The Internet of Things Architectural Reference Framework (IoT-ARF), designed within the Internet of Things – Architecture (IoT-A) project (Bauer et al., 2013), provides a functional overview of the various IoT components, as well as an information, a communication, and a trust, security and privacy models. However, no clear interaction among the components is proposed, as it is dependent on design decisions. The Software-defined Industrial Internet of Things architecture (SD-IIoT) (Wan et al., 2016) mainly provides clear communication protocols for data transmission from the physical layer to cloud environments. The architecture, though, does not provide guidance to develop IoT systems. The 8C architecture (Jiang, 2018) was proposed as an improved extension of the 5C architecture (Lee et al., 2015) for CPS for smart factories. The 3 added facets improved the emphasis on the horizontal integration, while 5C focuses mainly on the vertical integration. An example of a developed CPS system is presented but without structured instructions.

Li et al. (2018) performed a comparative analysis of most of the reference architectures mentioned above, including also others such as the Intelligent Manufacturing System Architecture (IMSA) (DKE, 2015; MIIT & SAC, 2015), developed by the Ministry of Industry and Information Technology of China (MIIT) and Standardization Administration of China (SAC), and the Industrial Value Chain Reference Architecture (IVRA) (Industrial Value Chain Initiative, 2016), developed by the Industrial Value Chain Initiative (IVI). They concluded that the construction of the reference architectures is based on three main principles; decomposition into multiple dimensions, focalization by focusing on specific smart manufacturing aspects and concepts and excluding others, and strategic consistency by embodying related national manufacturing strategies and initiatives, such as Industry 4.0. Bader et al. (2019) provide a structured analysis of existing reference frameworks, their classifications and the concerns they target. The work of Brouns (2019), not only positions numerous reference architectures in the fourth industrial revolution area but also proposes an integration framework for defining a roadmap towards smart manufacturing.

The current research adopts RAMI 4.0, as a widely adopted framework. Specific mapping of the designed artefacts onto the architectural model of Figure 23 are discussed in the corresponding sections.

2.1.5.2 Smart factory

The term *smart factory* has seen many definitions, as it is apparent from extensive literature studies (Osterrieder et al., 2020; Strozzi et al., 2017). There are references for equipping the machines with sensors to harness a "*continuous stream of data*" (Sjödin et al., 2018), machine-to-machine communication in networked systems for self-organization of processes and tasks (Tang et al., 2016), use of cloud systems for resources efficiency, use of autonomous devices (e.g., AGVs) for little or no human intervention, etc. Thus, the smart factory is a multi-aspected concept, similarly to the contexts of smart manufacturing and Industry 4.0, where it belongs (Figure 3). Osterrieder et al. (2020) propose a research model to approach the term from eight different pillars, listed in Table 3.

Table 3: Key pillars of the concept of "smart factory" (from research perspective) (Osterrieder et al., 2020).

Key pillar	Content
Decision making	Activities around data-based decision-making in manufacturing using different technologies, including visualization techniques, machine learning and AI. All kind of decisions in manufacturing, for instance design, scheduling, process planning and control are part of this research stream.
Cyber-physical systems	Developing concepts, models for assistant systems for operators, self-steering manufacturing systems and CPS, towards an autonomous running factory.
Data handling	Research activities deriving models and theories on how to exploit the potential of data with a focus on data generation, acquisition, mining and analysis. The objective is to provide data models towards a single source of truth and intelligent data exchange models.
IT infrastructure (hardware & software)	Discussion around the IT infrastructure of a factory to enable and foster a development towards a connected system. The field concerns with both horizontal and vertical data integration, thus requiring an interdisciplinary approach with the data handling field.
Digital transformation	Research on the transformational path of factories towards smart factories by including and focusing on the human perspective that comes along with this revolution.
Human machine interaction	Activities creating solutions for the co-automation, physical and digital assistant systems. Beside technological developments, the human perspective and role in autonomous smart factories is central within this stream. This pillar is solidly connected to decision-making and CPS.
Internet-of-Things	Accounts for the connectivity of elements and sensor technologies to increase transparency and traceability of (real-time) information about products and process states.
Cloud manufacturing and services	Split into a technology and business stream. The technology stream discusses cloud manufacturing architectures models and theories. The business stream highlights the business development perspective for smart factories, such as new operating models enabled by the digitalisation of factory capabilities.

A good way to describe what makes a factory smart, is to compare it to a traditional one. Wang et al. (2016) summarize such a comparison, presented in Table 4.

Table 4: Technical features of smart factory compared with the traditional factory (Wang et al., 2016).

Number	Traditional production system	Smart factory production system
1	**Limited and Predetermined Resources**. To build a fixed line for mass production of a special product type, the needed resources are carefully calculated, tailored, and configured to minimize resource redundancy.	**Diverse Resources**. To produce multiple types of small-lot products, more resources of different types should be able to coexist in the system.
2	**Fixed Routing**. The production line is fixed unless manually reconfigured by people with system power down.	**Dynamic Routing**. When switching between different types of products, the needed resources and the route to link these resources should be reconfigured automatically
3	**Shop Floor Control Network**. The field buses may be used to connect the controller with its slave stations. But communication among machines is not necessary.	**Comprehensive Connections**. The machines, products, information systems, and people are connected and interact with each other through the high-speed network infrastructure.
4	**Separated Layer**. The field devices are separated from the upper information systems.	**Deep Convergence**. The smart factory operates in a networked environment where the industrial wireless network and the cloud integrate all the physical artifacts and information systems to form the IoT and services.
5	**Independent Control**. Every machine is preprogrammed to perform the assigned functions. Any malfunction of single device will break the full line.	**Self-Organization**. The control function distributes to multiple entities. These smart entities negotiate with each other to organize themselves to cope with system dynamics.
6	**Isolated Information**. The machine may record its own process information. But this information is seldom used by others.	**Big Data**. The smart artifacts can produce massive data, the high bandwidth network can transfer them, and the cloud can process the big data.

In a broad sense, a smart factory embraces and integrates the recent Industry 4.0 technological advances in computer networks, data integration and analytics to bring transparency to manufacturing units (Lee, 2015). While most of the definitions give a purely intra-organizational concept, by considering a smart shop floor as the main realization of a smart factory (often referred to as *connected* factory), it should be noted that the concept may span beyond the boundaries of a manufacturing site or enterprise to embrace the extended supply chain (Davis et al., 2012). As Radziwon et al. (2014) define, a smart factory is not only "*a manufacturing solution that provides flexible and adaptive production processes that will solve problems arising on a production facility*", but also it could be seen as "*a perspective of collaboration between different industrial and nonindustrial partners, where the smartness comes from forming a dynamic organization*".

As the current research mostly focuses on the complexity (on various levels as illustrated in Figure 4) within factories, the term of smart factory is rather interpretated as a connected shop floor. To further understand the transition to smart production environments, key Industry 4.0 technologies have been explored (from various sources) and briefly listed below (the three broad categories implicitly follow the IoT technology stack of Wortmann & Flüchter (2015)):

- **<u>(Smart) Physical devices</u>**:
  - ***Robotics***, such as multi-axial arms with customized and reconfigurable end effectors (e.g., grippers) attached or collaborative robots (cobots) (Heyer, 2010; Pantano et al., 2021).
  - ***Automated guided vehicles (AGV),*** that transport material and products around a factory, for unmanned logistics (Le-Anh & de Koster, 2006).
  - ***Sensors*** that can measure (by responding to stimuli and generating processable outputs) physical data (e.g., temperature, vibration, light, etc.). They are embedded/attached in machines, on equipment or even human operators.
  - ***(Chipless) RFID tags*** (and their scanners) (Tedjini et al., 2010) that replace the optical barcodes or the quick response (QR) codes as they have larger data capacity, are adaptable and flexible in applications, are battery-less and low-powered (and thus more sustainable).
  - ***Augmentation devices***, like AR/VR glasses or head-mounted displays (Fraga-Lamas et al., 2018), to support operators in their daily tasks (Khan et al., 2011), which are getting more complex (Longo et al., 2017). Augmented reality is also used for remote maintenance or support (Palmarini et al., 2017).
  - ***Handheld, portable and wearable devices*** (Kadir & Broberg, 2020), e.g., smartwatches to provide information to the operator and allow for tracking of manufacturing tasks performed by humans, or wristbands to capture physiological signals such as heart rate, skin temperature and conduction, for monitoring physical and mental health conditions.
  - ***Data gloves*** (Fang et al., 2015). While joysticks, dials, and buttons have been commonly used as robot and machine input devices, data gloves provide the most natural and humanlike motion to interact with a device.
  - ***3D printers*** for additive manufacturing (Mpofu et al., 2014; Ngo et al., 2018; Shahrubudin et al., 2019), to create not only highly customized end products, but also parts that are used in manufacturing processes, e.g., a specific type of gripper to handle a new type of product.
  - ***(Inspection) Cameras***, for object detection, image recognition or quality inspection.

- **<u>Connectivity/Networking</u>**:
  - ***Wireless communication***, with protocols such as WiFi, Bluetooth, ZigBee, IPv6 over Low-power Wireless Personal Area Networks (6LoWPANS) (Kushalnagar et al., 2007), that allow connection of various devices in different network settings.
  - ***Industrial Internet of Things (IIoT)***, that brings together industrial internet (for real-time data availability and high reliability) and IoT (Sadiku et al., 2017; Thoben et al., 2017; Zhong et al., 2017).
  - ***Cloud/Fog/Edge networks***. The three available service models *Infrastructure as a Service (IaaS)*, *Platform as a Service (PaaS)*, *Software as a Service (SaaS)* (Zissis & Lekkas, 2012) can be used in the manufacturing domain as well (Xu, 2012). Depending on the industry

requirements/constraints (e.g., low latency), fog or edge computing is preferred (or combined) over cloud solutions (Caiza et al., 2020; Carvalho et al., 2019; Qi & Tao, 2019).

- **Computing applications**:
    - ***Big data*** algorithms and ***analytics*** to support and optimize decision making (Babiceanu & Seker, 2016; Lee et al., 2014; Ren et al., 2019).
    - ***Artificial Intelligence (AI)*** for assisting in maintenance activities, in image-based object recognition (Kim et al., 2021; Rai et al., 2021; Tran, 2021), in augmented reality tasks by users (Sahu et al., 2020), and other data-intensive activities.
    - ***Virtual computing***, to enable concepts of *virtual factory* (Terkaj et al., 2015), *digital factory* (Gregor et al., 2015), and *digital twin* (Cimino et al., 2019; Tao et al., 2019), for simulation, testing and evaluation of physical systems.

It is noteworthy to highlight that while most of the recent Industry 4.0 developments encountered in production environments focus on making robotics and machines more autonomous and intelligent, there are also ones that aim at supporting humans in their daily tasks and improving their well-being at work, enabling the concept of "*Operator 4.0"* (Longo et al., 2017; Romero et al., 2016).

2.1.6 Challenges and problems in smart manufacturing

Transforming into a smart factory is a challenging endeavor for manufacturers. The adoption of Industry 4.0 advancements, while promising positive consequences, comes with collateral effects and costs, as the higher the intelligence and automation in a factory, the higher the complexity (Qin et al., 2016; Sjödin et al., 2018). Complexity, scoped either to parts, products, systems, or system of systems (Mourtzis et al., 2019), should not be confused with complicatedness. As Elmaraghy et al. (2012) distinguish in their spectrum of process complexity (ranging from simple to chaotic systems), "*a complicated system could refer to a system having many parts, making it somewhat harder to understand, perhaps by virtue of its size, whereas complex refers to a system containing uncertainty during the development process or intrinsically in its design, the outcome not being fully predictable or controlled*". Thus, complexity exists where is uncertainty, which together with variability, are concepts embedded in the fundamental nature of manufacturing (Hon, 2005). Especially in the smart manufacturing era, where the plethora of new technologies and concepts might cause a lack of common understanding (Sjödin et al., 2018), product variety is rather the norm, and distributed global markets increase volatility (Mourtzis et al., 2018), manufacturing systems are getting more and more complex.

In research literature, the concept of complexity is reviewed from three perspectives (Elmaraghy et al., 2012): (i) complexity of engineering design and the product development process, (ii) complexity of manufacturing processes and systems, and (iii) complexity of the global supply chain and managing the entire business, as well as their intersections. As the current research rather focuses on how the manufacturing operations are being transformed towards realization of the smart factory concept, the second perspective is taken. That means that the focus is not on how to design a complex product (first perspective), but how the processes and systems are organized to produce it. Also, as the focus is on the operations within the boundaries of a factory (as mentioned in Section 2.1.5.2), the broader scope of the global supply chain (third perspective) is not considered. Though, as these perspectives have intersecting scopes, references from the two excluded perspectives are taken into account as well (for example, how a market event can affect a running manufacturing process).

This section discusses identified causes in literature that lead to the general complexity issue in manufacturing operations (see *Problem statement* in Section 1.3). Tackling complexity in engineering is achieved with the use of methods and tools to transform the problem from complex to manageable and controlled (Elmaraghy et al., 2012). Thus, the general problem is decomposed into smaller problems, by grouping relevant causes together, which will then allow for better design and implementation of solutions. Four general problems have been identified, namely *complex production scenarios*, *rise of exceptions*, *resource under-utilization* and *integration complexity*, which have been summarized in Section 1.3 and illustrated in Figure 4. While the resource under-utilization problem has been covered in Erasmus (2019), and thus not further discussed here, the rest problems are investigated below.

Of course, the following problems are not the only ones faced by enterprises which strive to enter smart manufacturing, but it is the specific angle of the identified problem under research that puts the focus on those. General challenges and barriers have been extensively studied (Horváth & Szabó, 2019; Moeuf et al., 2017; Raj et al., 2020), where aspects such as lack of (financial and infrastructure) resources, organizational resistance to change, cyber-security risks, or lack of digital skills are proved to be hindering the adoption of Industry 4.0 technologies and concepts, especially by SMEs (Devos et al., 2014; Mittal et al., 2018; Schroeder, 2016; Stentoft et al., 2019).

2.1.6.1 Complex production scenarios

In recent years, the demand for customized and personalized products, together with the fierce competition among manufacturers in global markets, has increased the product variety offerings. When the products are complicated, like a truck or an aircraft, the variety is distinguished in various levels. ElMaraghy (2009) provides a product variety hierarchy in automotive industry, consisting of eight levels: i) part features, ii) parts/components, iii) parts family, iv) product modules or sub-assemblies, v) products, vi) products families, vii) products platform, and viii) products portfolios. With hundreds, or thousands, or even millions[18] of parts for a single product, the combinations can get extremely high. For example, the number of possible vehicle variations in the BMW 7 series alone could reach 10^{17} (Hu et al., 2008), while the Dutch truck manufacturer DAF Trucks (see Figure 2 for their online configurator) claims that each truck is unique[19]. Inevitably, product variety results in production variety (Johansson et al., 2016) and complexity, as it takes hundreds of manufacturing and assembly steps to produce all the variants (ElMaraghy et al., 2013). Considering, also, that most products nowadays incorporate not only mechanical and electrical components but also software and human–machine interfaces (HMI), the complexity of production operations is increased (ElMaraghy et al., 2013).

[18] https://edition.cnn.com/travel/article/airbus-a380-parts-together/index.html
[19] https://www.daf.com/en/news-and-media/news-articles/global/2021/q3/daf-trucks-flanders-3000000-axles-in-50-years

High product customization and personalization depends on the point of customer involvement (Duray et al., 2000), also referred to as customer decoupling point (Rudberg & Wikner, 2004) or order penetration point (Olhager, 2003). Customer can be injected in any of the four main stages of the production cycle, i.e., design, fabrication, assembly and distribution, with pure customization (or personalization) happening when the customer is involved at the design of product specifications (Lampel & Mintzberg, 1996). This decision determines accordingly the engineering strategies. The most established ones are the build/make-to-stock (BTS/MTS), assemble-to-order (ATO), make-to-order (MTO), engineer-to-order (ETO) or design-to-order (DTO) (Mahdavi & Olsen, 2017). In high-customization environments, enterprises employ upstream decoupling points, i.e., involve the customer as early as possible, and thus MTO or DTO are the preferred strategies (Um et al., 2017). This is typically related with low volumes and high mix production, which requires frequent and rapid changes in activities and use of machinery and equipment.

Achieving lean manufacturing in "high mix – low volume" environments is a complex and still under research topic (Tomašević et al., 2021). The recent Industry 4.0 technologies can enable manufacturing and assembly systems to switch fast between variants of parts and products (Johansen et al., 2021). Collaborative robots are employed in (final) assembly to assist humans (Karaulova et al., 2019). There are various levels of corporation between a human worker and a robot (Bauer et al., 2016), as shown in Figure 24. The different interaction styles, though, require adaptation of current working styles. Similarly, the introduction of an AGV to assist with transportation tasks might result not only in restructuring of the physical layout (e.g., by building navigation paths), but also the sequence of activities. Considering also safety requirements to prevent hazards for humans, the production operations increase the level of complexity, with extra parameters and unexpected events that have to be taken into account (Gualtieri et al., 2020).

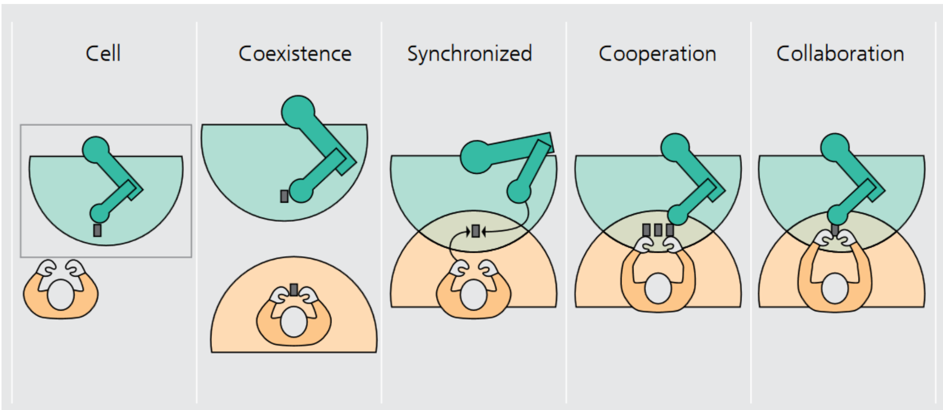


Figure 24: Human-robot collaboration levels (Bauer et al., 2016).

2.1.6.2 Rise of exceptions

Manufacturing products, processes and production systems are being challenged by evolving forces, including high demand turbulence, introduction of new regulations, pressure on costs, demand for shorter delivery times and (increasing number of) new technologies (Tolio et al., 2010). These forces create complex dynamics in the domain, as illustrated in Figure 25. The dynamicity and volatility, arising either within an organization, or endogenous and exogenous to its corresponding supply chain (Nitsche & Straube, 2020), often generate unplanned events, i.e., exceptions, that disturb the manufacturing processes and require quick actions (Block et al., 2018).

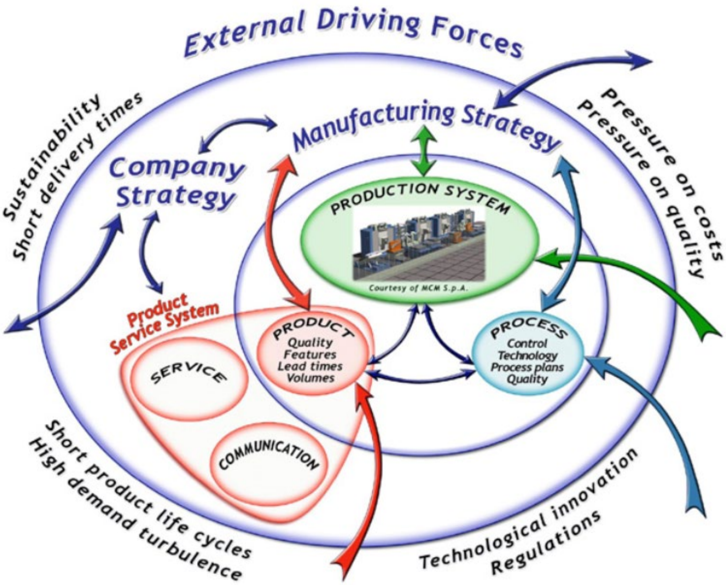


Figure 25: Complex dynamics in manufacturing (Tolio et al., 2010).

Involving customers, e.g., through ETO or DTO strategies, might result in change of requirements that have to be reflected in production systems. An order alteration or cancellation while the product is under processing will impose re-arrangement of activities and resources. Similarly, late supply deliveries might result in re-scheduling of production to avoid waste of time or even trigger production in another facility where the stock is available. It is crucial then that such deviations from the normal flows are captured and transferred as soon as possible to the involved systems.

Within the boundaries of the factory, uncertainties manifest themselves in the form of equipment breakdowns or malfunctions, tasks rejects and reworks, labor absenteeism and turnovers, material mishandling, etc. (Sawhney, 2006). With the use of various new technologies, some of them still not on mature level, the exceptions are likely to be more frequent. While fully automated devices can have self-recovery mechanisms, systems that still keep humans in the loop need to provide the right and in understandable way information for any manual corrective actions (Gorecky et al., 2014). Especially in production scenarios with high degree of variety and configurations, current manufacturing systems are rigid and static, as each error might require different handling (Keddis et al., 2016). Moreover, in distributed and decentralized systems, error handling requires well defined actions, as more than one resources might have to be restarted (Farooqui et al., 2016).

2.1.6.3 Integration complexity

Manufacturing involves a lot of functions and activities, including fabrication, assembly, packaging, sales, maintenance, quality control, IT support and more, performed by various stakeholders such as producers, suppliers, logistics experts, system integrators, operators, etc. The functions, grouped in various levels as shown in Figure 7, are supported by corresponding information systems, as shown in Figure 21. Integration of systems, especially across functional levels is a challenge. Unver (2013) talks about disconnection between enterprise and shop floor systems. Lanza et al. (2019) and Orzes et al. (2020) have discovered data silos in manufacturing systems and applications. Scenarios of reacting on external triggers (like the ones mentioned in Section 2.1.6.2) are hard to address if, for instance, ERP and MES systems are not well integrated. Respectively, shop floor data are not efficiently exploited if they are not propagated to enterprise systems for better decisions. Considering, also, the introduction of new technologies, the integration is getting more challenging. As Frank et al. (2019) summarize in their categorization of Industry 4.0 technologies, shown in Figure 26, the higher the automation and flexibilization desired, the higher the complexity level to integrate and implement the technologies that promise those desires.

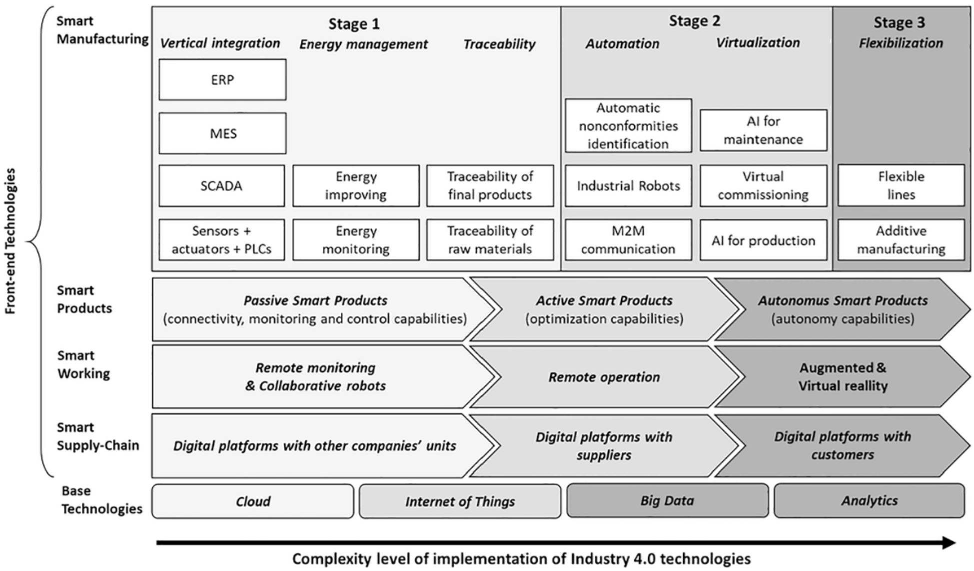


Figure 26: Increasing complexity of implementation of Industry 4.0 technologies (Frank et al., 2019).

The lack of integration of technology platforms is a challenge for adopting Industry 4.0 initiatives (Luthra & Mangla, 2018), but CPS can facilitate such integration (Wang et al., 2015). However, a CPS is inherently a complex system as its elements are heterogeneous technologies of various forms (as the ones introduced in Section 2.1.5.2) (Napoleone et al., 2020). Moreover, as many companies start the journey towards smart manufacturing in an incremental way by building solutions on top of legacy systems and not from a greenfield landscape (Tabim et al., 2021), the technologies heterogeneity challenge is greater towards achieving the vertical and horizontal integration that the new paradigm requires.

Seamless integration of technologies, systems and the operations those support is also hindered by the fact that smart factory technologies are applied in a piecemeal approach, resulting in isolated, fragmented and uncoordinated operations from a broader perspective (Strozzi et al., 2017). Deploying smart solutions on separate production lines might result in local process improvements, but the full potential is not being leveraged across all production lines or the entire factory for end-to-end process management (Schuh et al., 2020).

2.1.7 Approaches to face challenges and problems in smart manufacturing

The operations complexity problem in the new manufacturing era is multi-dimensional and broad. Various paradigms have been proposed and applied in response to the challenges discussed in Section 2.1.6. Recent developments are discussed in this section to discover the state of the art in manufacturing operations management.

2.1.7.1 Model-driven paradigm

The model-driven (MD) paradigm orientates towards the use of models to facilitate the specification of a system, its parts, its structure and behavior. MD as a prefix (often referred to as model-based) is an umbrella term to indicate approaches like MD engineering (MDE), MD development (MDD), MD architecture (MDA), MD software engineering (MDSE). MDE (Schmidt, 2006) is rather a software engineering approach that considers models not just as documentation artefacts, but also central artefacts, from which software systems can be created and automatically executed (Rodrigues Da Silva, 2015). MDE still faces great challenges and adoption barriers. Bucchiarone et al. (2020) classifie the challenges on foundation, domain, tool and implementation, social and community levels, while Mussbacher et al. (2014) stresses the lack of (industrial) evidence of benefits and consequently its limited widespread adoption in industry. However, as the paradigm allows coping with the complexity of reality by abstracting the relevant aspects of a system or an application into corresponding models (Bucchiarone et al., 2020), its principles and methods are useful in building solutions for smart manufacturing.

Jeschke et al. (2017) consider the concept of digital factory as a comprehensive model of the real factory that can be used for communication, simulation and optimization during its life cycle. Combined with PLM models in a network of digital models, the physical world of products and their production are represented in an IIoT setting. Cadavid et al. (2015) apply MDE techniques to conceive the concept of smart factory with the following six characteristics: i) use of specialized model libraries for manufacturing processes, ii) a specific architectural framework composed of a dedicated architecture viewpoint description, iii) automated and context-aware generation of user interfaces, iv) tailoring of manufacturing processes, v) traceability management and vi) model-based simulation. Similarly, Pérez et al. (2015) see MDE as a fitting approach for building cyber-physical production systems (CPPS) as it, with respect to methods and techniques, offers: i) separation of concerns through domain modeling, promoting system descriptions from different points of view, ii) domain model mappings through model-to-model transformations, supporting consistency analysis and iii) automatic generation of configuration files and code through model-to-text transformations. Moreover, Vještica et al. (2021) use modeling for representation of production processes, while Calvary et al. (2014) discuss the interaction between human-computer interface (HCI) engineering and MDE to build and transform more efficient user interface (UI) models.

2.1.7.2 Event-driven process management

Event-driven approaches are the response to the need for reaction on changing situations, through immediate identification of events and their correlation. Be it business and market events, or low-level machines and sensors data, organizations have to catch and integrate them, refine them into useful information and make timely decisions (preferably in an automated way). The paradigm has been widely adopted in various industries, like for example in transportations and logistics, where real-time information (e.g., a strike or bad weather conditions) can be used to adjust the planning of freight transportation (Baumgraß et al., 2016).

In smart manufacturing, the plethora of events and the enabling technology, make the paradigm a good fit to address the challenges of responsiveness and situational awareness. Yao et al. (2018) propose a radio-frequency identification (RFID) event-driven integrated production planning and control framework to address the growing need for rapid responsiveness to customers' orders (in MTO production environment). Theorin et al. (2017) present an event-driven manufacturing information system architecture, in a service-oriented architecture (SOA) fashion, to enable flexible factory integration and data utilization. It offers control of both low-level applications and aggregation of higher-level information, such as key performance indicators (KPIs).

Event-driven approaches are often combined with the BPM paradigm, aiming at providing support on operational level. Krumeich et al. (2014) discusse the application of complex event processing (CEP) in the context of BPM. An event-driven business process management approach, resulting in operational transparency, builds the foundation for (near) real-time reactions. Similarly, Estruch & Heredia Álvaro (2012) introduce event-driven manufacturing process management by combining BPM with CEP to model the logic of complex events in manufacturing processes.

2.1.7.3 Agents-based manufacturing

Towards distributed, intelligent and autonomous systems, agent technology is a promising paradigm. The term agent refers to an entity that is capable to take action continuously and autonomously to meet its design objectives, in dynamic environments where often other entities and processes exist (Adeyeri et al., 2015). The term holon is also used to denote an autonomous and cooperative building block of a system, which consists of an information processing part (software component) and a physical processing part (hardware component) (Colombo et al., 2006). The premise of the paradigm is that since each agent has computational and decision-making capabilities, agent-based (or multi-agents) systems will react faster in changes (Mařík & McFarlane, 2005).

Agent-based systems have already been applied in manufacturing (Leitão, 2009; Monostori et al., 2006). Most of the approaches are focused on production scheduling optimization (Rolón & Martínez, 2012) or self-organizing manufacturing systems (Barbosa et al., 2015; Park & Tran, 2012; Schild & Bussmann, 2007). These approaches typically require homogenous manufacturing systems, occupied by robots that can negotiate with each other. Alternatively, these approaches opt for software deployment only, where the physical agents are represented by software agents who negotiate a course of action. The course of action is then translated to a production schedule, which is deployed by a different production control information system.

Giving a process-oriented perspective in the communication of agents, subject-oriented business process management (S-BPM) (Fleischmann et al., 2015) has been proposed for smart manufacturing (Kannengiesser et al., 2015). While practical demonstrations and evaluations have been performed (Neubauer & Stary, 2016), the uptake of the approach is rather limited despite the promises, possibly due to the lack of interest of the used modeling notation.

2.1.7.4 BPM approaches

BPM, as a successful paradigm in various domains, has already seen interest in (smart) manufacturing (Castro & Teixeira, 2021; Janiesch et al., 2017) with positive effects (Gažová et al., 2022). There are efforts focusing both on the modeling and the execution support of manufacturing processes. For instance, Abouzid & Saidi (2019), Prades et al. (2013) and Zor et al. (2010, 2011) have used and extended BPMN, the de-facto standard for business process modeling (Chinosi & Trombetta, 2012; Decker & Barros, 2008). Similarly, Meyer et al. (2013, 2015) and Petrasch & Hentschke (2016) focus on modeling IoT-aware processes. All of these approaches though, do not provide execution support.

Schönig et al. (2020) provide an integrated approach that exploits IoT on top of BPM concepts. The focus is on (mobile) task assistance and decision support systems, without though discussion on resource allocation, exception handling or process monitoring. A more complete application of BPM in smart manufacturing has been provided by Pauker et al. (2018), with tools to design manufacturing processes and a process engine to enact those. However, aspects such as safety or situational awareness are not considered.

2.2 Practical cases

Having ensured scientific rigor by studying the literature for the current state-of-the art of the aspects of manufacturing that this research deals with, practical relevance (in accordance to the DSR framework of Figure 13) is ensured by examining practical cases in real environments in manufacturing industry. These cases are in the context of three research and innovation European projects, namely HORSE, EIT OEDIPUS and SHOP4CF, in which the author of this thesis has been actively involved. The projects are briefly introduced in Section 2.2.1, while the subsequent three sub sections present the practical cases (the cases presented in this thesis have been selected as representative examples for analysis and application scenarios, out of a substantial set of cases, summarized in Appendix B).

The scenarios of the practical cases are presented with the following two purposes:

1. To validate the problems identified from scientific literature, as discussed in Section 2.1.6.
2. To serve as testbed to realize and evaluate the artefacts of this research in real-world operational environments, as discussed in Chapter 7.

2.2.1 Involved EU projects

A brief introduction of the three EU projects is given below.

2.2.1.1 HORSE

HORSE[20] was a Research and Innovation Project in the EU Horizon 2020 program, running from 2015 to 2020, as a collaboration of fifteen organizations across Europe, among which academic and applied research institutions technology providers and manufacturing enterprises. The goal of the project was to make a significant advancement in the industrial use of smart manufacturing by developing an integrated framework (Grefen & Boultadakis, 2021) that extends and unifies several state-of-the-art technologies, including smart robotics, business process management technology, as well as the development of software technology components to implement this framework in practice. The project focused on SMEs, as these enterprises face the greatest challenges in the adoption of smart manufacturing technology. Also, it focused on the discrete manufacturing domain, i.e., on companies that manufacture individual or batches of individual products, as this domain faces the most pressure from the mass customization and personalization trends. Apart from initial three factories, which served as pilots, seven more experiments managed to integrate and use the system in their industrial environments (Traganos et al., 2021).

2.2.1.2 EIT OEDIPUS

OEDIPUS[21] (Operate European Digital Industry with Products and Services) was a research and innovation program funded by the European Institute of Technology under its EIT Digital overall program. OEDIPUS covered several use cases between 2017 and 2019. Within these, the author of this thesis was involved in the Print 4.0 use case, which focused on the application of flexible end-to-end process orchestration of manufacturing processes in smart printing factories (Traganos et al., 2020).

2.2.1.3 SHOP4CF

SHOP4CF[22] (Smart Human Oriented Platform for Connected Factories) is an EU-funded project within the eighth framework program Horizon 2020, running from 2020 to 2023. SHOP4CF aims to create a unique infrastructure for the convenient deployment of human-centric industrial applications. In the project, twenty partners develop a comprehensive software platform containing a wide range of components that cover a broad spectrum of industrial requirements, especially in the context of flexible and data-rich manufacturing. SHOP4CF aims to find the right balance between cost-effective automation of repetitive tasks and involve the human workers in areas such as adaptability, creativity and agility where they create the greatest added value. In doing so, the project pursues a highly connected factory model to reap the benefits of all data generated within a factory.

[20] http://www.horse-project.eu/ - The full title of the project is “Smart Integrated Robotics System for SMEs Controlled by Internet of Things based on Dynamic Manufacturing Processes”, but a shorter name of the project was chosen for pragmatic reasons, referring to the analogy between a robot in modern manufacturing and a horse in old times.

[21] https://www.eitdigital.eu/innovation-factory/digital-industry/oedipus/ - Operate European Digital Industry with Products and Services; OEDIPUS Fact Sheet; EIT Digital: Brussels, Belgium, 2017.

[22] https://shop4cf.eu/

2.2.2 Thomas Regout International (TRI)

Thomas Regout International[23] (TRI), a pilot partner in the HORSE project, is a Dutch manufacturer, based in Maastricht, the Netherlands, specialized in the design and production of customized telescopic slides (examples shown in Figure 27). The telescopic slides provide horizontal and vertical movement, e.g., those used in drawers and cabinets to protrude and extract, with several industrial equipment applications including heavy machinery, automotive and aerospace. Although a typical slide consists out of only five parts (three metal profiles and two ball-bearing cages), the high customization of these parts results in approximately 900 product variations.

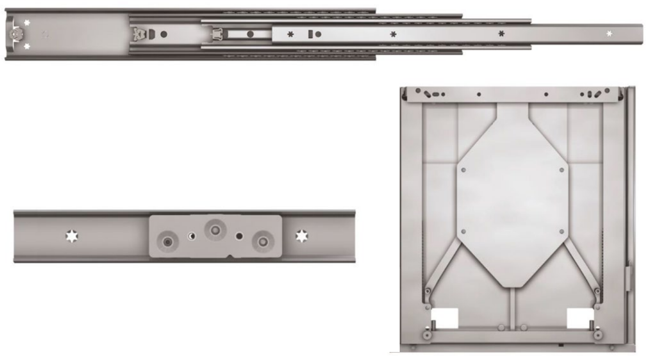


Figure 27: TRI's telescopic slides and vertical balance systems.

TRI, with its high-quality products and short delivery times of customized, small batches, is a global leader in their market. However, the company faces problems in their production and they aim to transition into smart manufacturing to keep and enhance its competitive advantages.

2.2.2.1 Production layout

TRI can be classified as a small batch producer (Woodward, 1965), with its factory considered as a reconfigurable manufacturing system. The high-level production process, visualized in Figure 28, involves three main production phases in respective production areas:

- Cold forming of steel coil into profiles, stamping and welding (P1).
- Surface treatment of the steel profiles (P2).
- Final assembly of slides (profiles and ball-bearings) (P3).

Additionally, a tool preparation production area precedes P1, for assembling tools for the cutting, stamping and bending operations.

[23] https://www.thomasregout-telescopicslides.com/home

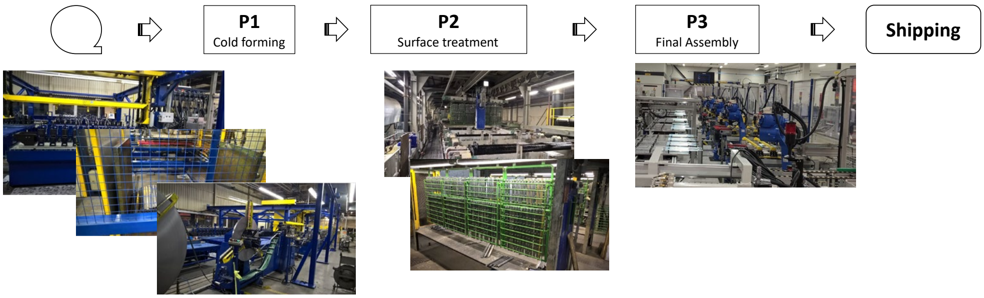


Figure 28: High-level production process at TRI, consisting of three main production phases.

P1 shapes coils of steel sheets into the main three or four profile types. It consists of the following steps:

- Load steel coil into the machine.
- Cut the steel coil into profiles lengths.
- Stamp holes and bend lips according to product and production requirements.
- Bend lips according to assembly requirements.
- Deburr the profiles to remove unwanted material.

Approximately 60% of profiles are processed on an automated production line, called Profistans, consisting of a computer-operated crane for switching between coils and a series of hydraulic presses. The remaining 40% of profiles are either too small or large for the automated line, thus necessitating manual execution at three human-operated production lines, specialized for different product dimensions.

P2 improves the corrosions resistance of the profiles through electro-galvanic zinc plating. It consists of the following steps:

- Load profiles on racks.
- Automatic transportation of racks to galvanic process.
- Electro-galvanic zinc plating of profiles (on the racks).
- Automatic transportation of racks to unloading position.
- Unload profiles from the rack.

Loading and unloading profiles on/from racks are currently performed completely manually by operators, a rather heavy-loaded task. Automated (un)loading is difficulty with current machinery as the profiles are hooked on through small holes.

P3 assembles the prepared steel profiles and ball-bearings into the finished telescopic slides. It consists of the following steps:

- Assembly of telescopic slides (final product).
- Quality control of final product.
- Packaging for warehousing and distribution.

Three production lines perform the final assembly in P3. A fully automated assembly line, consisting of eight robots, assembles about 60% of the production volume. However, the parts are loaded into and unloaded from this line by human operators. A semi-automated assembly line handles at about 35% of the volume. A series of specialized work cells compose a manually assembly line for the remaining 5% of the production volume.

Lastly, three storage zones are also located in the factory, two as buffers between P1-P2 and P2-P3, and one warehouse zone. Figure 29 shows the physical hierarchy of TRI, with the blue-highlighted work cells to indicate the focus of the application scenarios within HORSE project.

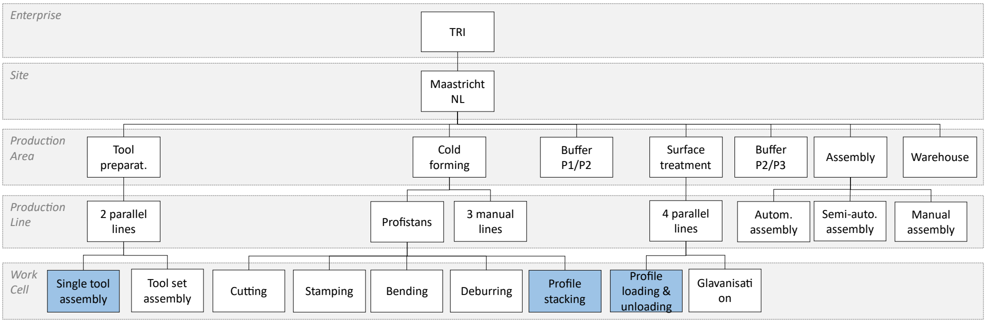


Figure 29: Physical hierarchy of TRI, according to IEC62264-1 standard, with the three blue-highlighted work cells as the focus of the application scenarios within HORSE project.

2.2.2.2 Manufacturing process

The operations in tool preparation phase and the three main production phases are described with BPMN, as a high-level process model, shown in Figure 30. Each of the four phases has been modeled as separate process (swim pools). This respects the physical separation, but from a functional perspective it indicates whether the phases are well-connected or not. Indeed, while a production order has to go through all the phases in a sequential way, in the current situation there is no automated production flow. Semi-finished products from P1 are temporarily stored in a buffer zone, until retrieved for further processing in P2.

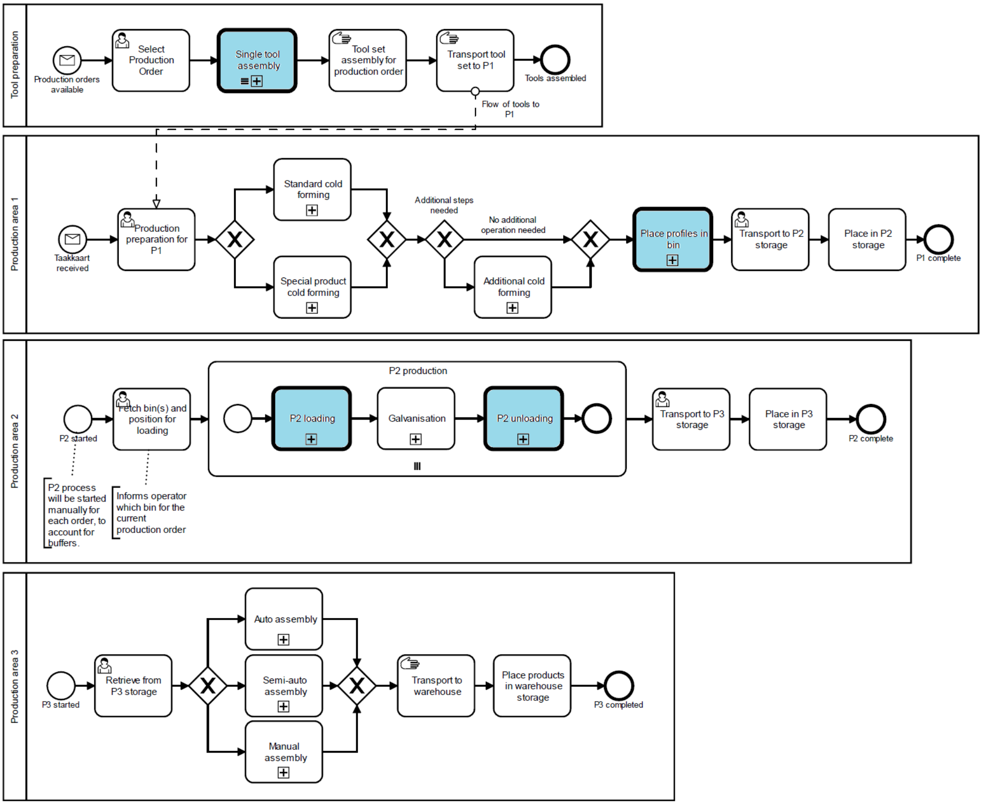


Figure 30: TRI current production process, with a tool preparation phase and three main production phases, modelled as high-level BPMN processes.The bue-highlighted subprocesses are the main focus of Industry 4.0 interventions.

2.2.2.3 Support systems

Figure 31gives an overview of the support systems of the production, in accordance to Figure 21. A cross-functional ERP system (Microsoft Dynamics) processes customer orders and manage company resources. It also generates production orders, whose scheduling is performed by a proprietary production planning system. The latter generates work instructions, printed on paper and moving along the production areas.

TRI has a achieved a good level of automation at the physical level (in disparate control fashion though), but not the same at a system support level, as can be seen at the overview with the disconnection of Levels 3 and 2. Instead, it solely relies on verbal communication among personnel and the printed instructions that can be easily lost around the factory.

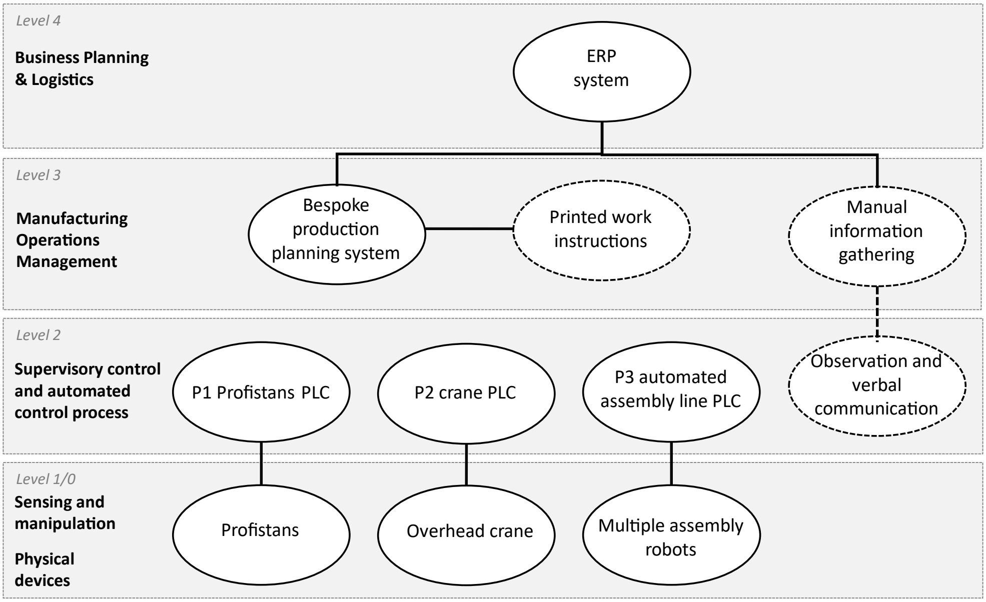


Figure 31: Overview of hardware, software and human support of production at TRI, positioned across the functional hierachy levels of IEC62264-1 standard.

2.2.2.4 Identified problems

The type of production and the way production is organized at TRI raise issues at operations level from various aspects.

Operating in high mix-low-volume environments requires frequent changes of machines and tools to manufacture the different product types. This is takes time and is error prone. Unplanned down time is relatively high. While this can be attributed to many causes, as illustrated with the fishbone diagram of Figure 32, it results in under-utilization of resources.

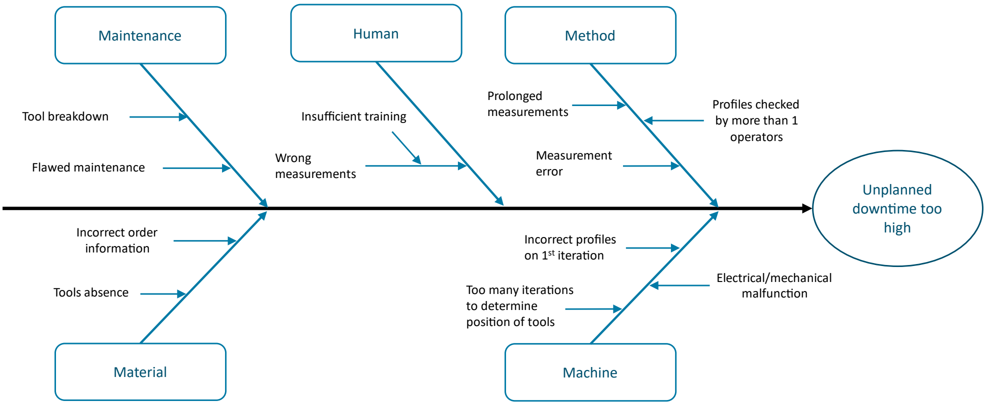


Figure 32: Identified causes of high unplanned downtimes, presented as an Ishikawa (fishbone) diagram.

At P1, every different type of product requires different tools to be produced. The high product variety makes infeasible to have the tools readily available. Instead, tools are assembled on demand from modular parts. Tool assembly is a precise and complex task performed by experienced operators. The heavy reliance on experts, not only can increase the costs (and consequently the product price) but also causes problems in production from a time perspective. When tool experts are not available, P1 is delayed.

At P2, the heavy tasks of loading and unloading profiles on racks place a lot of physical stress on human operators. While challenging, both technically and financially, a robotic solution should be found to alleviate operators.

The automated hanging of profiles on racks for galvanization is also hindered by the way profiles are organized between P1 and P2. Profiles are dumped unstructured in a crate and placed in buffer zone. The operator has to manually lift them and place them in order for the next operation, a repetitive and tedious task.

The disconnected production and storage areas does not allow for continuous production flow. Apart from any physical limitations, this is also caused by a missing process management information system to provide control and overview of operations. Activities are organized through verbal communication and printed instructions. Without digitalization of work instructions, there is no transparent vertical integration across control levels.

While recognizing the challenges of automating some of its human knowledge and skill dependent operations, TRI aims to transform into a smart factory. Through innovative robotic solutions and a process orchestration system, the company hopes to address the identified problems.

2.2.2.5 Intervention scenario(s)

TRI identified three scenarios for intervention, which are discussed in this section to demonstrate the intended use of process management to support production operations. The current way of operating and the desired changes of each scenario are discussed individually in the following three sub-sections.

2.2.2.5.1 Single tool assembly

Tool assembly is performed by a few highly skilled and experienced employees. It takes up to two years of on-the-job training to reach the base level of tool assembly competence. Unavailability of experts causes delays, while incorrect tools can cause loss of production, rework and eventually extra costs. The process is linear and fully manual. Operators select one of the available printed productions orders, which requires a set of tools to be assembled. They, then, assemble each single tool one-by-one, as shown in the top process model of Figure 30. The single tool assembly subprocess is shown in Figure 33.

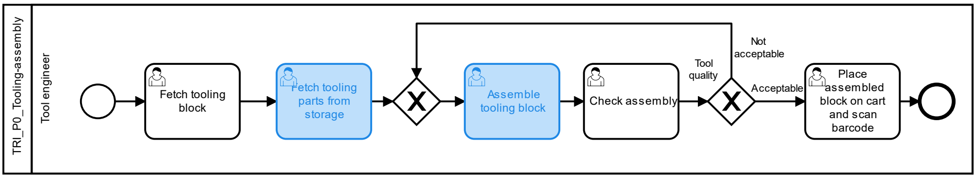


Figure 33: Single tool assembly process at TRI (current), with the blue-highlighted activities as the main focus of Industry 4.0 interventions.

The process will undergo the following three changes through the introduction of modern technologies:

1. The task "fetch tooling parts from storage" will be performed by a mobile robot, with a robotic arm mounted on it, such that the assembly process does not need to be interrupted and less errors are made.
2. The task "assemble tooling block" will be supported by augmented reality, guiding inexperienced operators through the steps of tool assembly.
3. Introduction of process management technology to orchestrate the activities of the human and robot.

2.2.2.5.2 Profile stacking

Intermediate products after P1 and P2 areas are placed into containers for transportation to storage zones before the next production area. The three-steps process is shown in Figure 34.

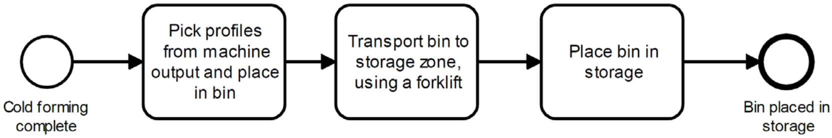


Figure 34: Profile stacking and transportation process at TRI (current).

The process will undergo the following two transformative changes:

1. The task "pick profiles from machine and place in bin" will be performed by two robots. The first will pick up the profiles and place them on a conveyor belt, then the second will pick up the profiles form the conveyor belt and place them, in an orderly fashion, in the bin.
2. The remaining two tasks (transport bin to storage zone and place bin in storage) will be performed by an autonomous guided vehicle that is specifically designed to transport bins.

2.2.2.5.3 Loading and unloading of profiles

Profiles are hooked onto racks, the racks then are moved through galvanization baths, and upon exiting from there, they are emptied by unloading the profiles onto bins. The scenario is split into two process models the automated galvanization process decouples them both from a physical and time perspective. Also, the loading and unloading is performed rather ad-hoc based on the available spots on the racks. Figure 35 shows the loading process. All tasks are completely manual except the last task performed automatically by the crane.

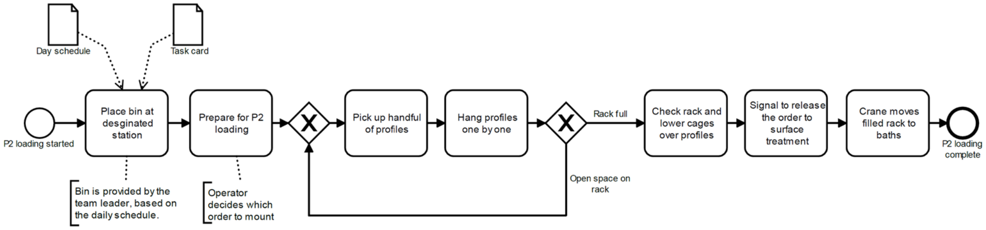


Figure 35: Loading of profiles onto racks for galvanization treatment at P2 phase at TRI (current).

Figure 36 shows the unloading process. All tasks are manual except the first one.

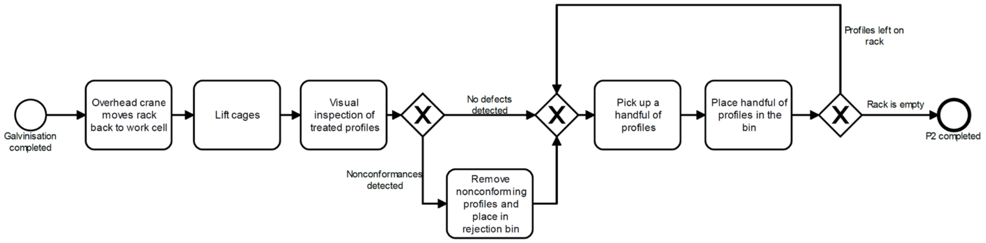


Figure 36: Unloading of profiles from racks after galvanization treatment at P2 phase at TRI (current).

Both processes will undergo similar changes:

1. A conveyor belt is introduced, on which a robotic arm will place the orderly stacked profiles from the bin (from the second scenario).
2. A second robot picks a set of profiles from the conveyor belt and hooks it onto the rack.
3. Opposite tasks will be performed for the unloading part.

2.2.3 Canon Production Printing (CPP)

Canon Production Printing[24] (CPP) (formerly known as Océ), a partner in the EIT OEDIPUS project, is a global leader in consumer and professional imaging. They provide technologies, products and services for their main markets in printing, with customers ranging from local creative studios to global blue-chip multinationals. In digital media printing (e.g., books, brochures, magazines), they offer industrial printers, like the ones shown in Figure 37.

[24] https://cpp.canon/

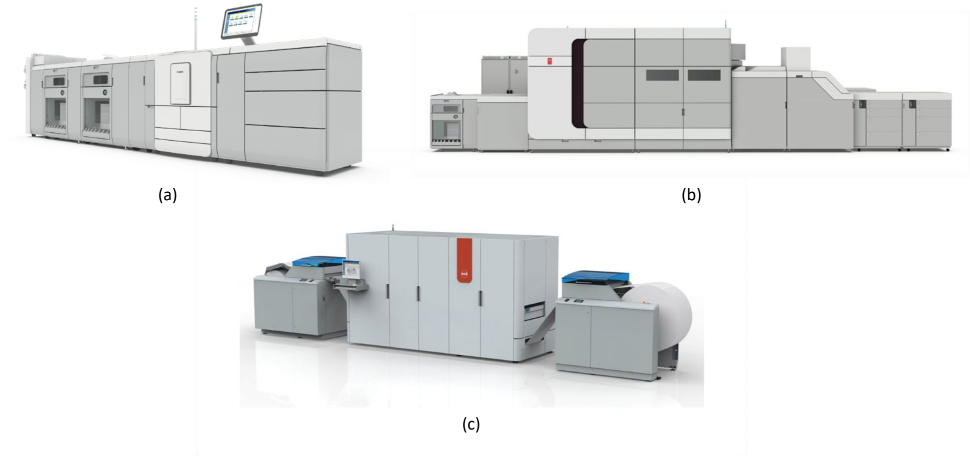


Figure 37: CPP industrial printers (a) monochrome production printer for light and mid production VarioPrint 140[25], (b) high-volume colour inkjet sheetfed press VarioPrint i300[26], (c) continuous feed inkjet press ColorStream 3000Z[27].

Production activities in a print shop include, apart from the printing by the printers, media (e.g., paper) loading and unloading, material binding, transportation of (semi-)finished products, etc. The volume of production, the variety of products requested, and the number of equipment can cause complexity and inefficiency in production printing. In response to customer demands for facilities and operations optimization, 24x7 production printing, and less human errors, CPP aims to realize the concept of factorization of printing, where all activities on a shop floor will be fully automated. This of course requires not only introduction of robotic solutions (e.g., robotic grippers or transportation AGVs) but also system support for process orchestration.

2.2.3.1 Production layout

A print shop, like the one shown in Figure 38, is not a typical manufacturing environment, and as such, it cannot easily be structured according to the physical hierarchy of IEC62264-1 standard. The notion of production lines and work cells does not really apply. Every print shop organizes its facilities according to the equipment they possess and the physical layout of their shop. Equipment can be organized based on the type of printers (e.g., arrange all monochrome printers in the same area) or based on the sequence of production activities. In the latter case, the layout can resemble a production line. Also, there are usually storage zones to keep (semi-finished) products.

[25] https://cpp.canon/products/varioprint-140-series/
[26] https://cpp.canon/products/varioprint-i-series/
[27] https://cpp.canon/products/colorstream-3000z/

Figure 38: An example of a print shop, with different types of printers, storage places, and moving carts[28].

Apart from printers, other typical machinery used in production printing are forklifts or trolleys to move media, binding machines to bind together a printed cover and a bookblock, and cutting machines (guillotines) to trim the printed media in the correct size. Examples of the last two types of machines, called finishing machines, are shown in Figure 39.

[28] Courtesy of https://www.bookmobile.com/book-production/bookmobile-thirty-three-years-old-st-patricks-day-origin-story/, used under fair use policy.

(a)

(b)

Figure 39: Examples of finishing machines at a print shop (a) binding machine[29], (b) cutting machine (guillotine)[30].

2.2.3.2 Production process

The media print production flow comprises three main stages (Kipphan, 2001); the prepress stage (e.g., pre-flight, vector conversion, imposition, trapping), the press stage (the actual printing), and the postpress stage (e.g., binding, stapling, trimming), summarized in Figure 40.

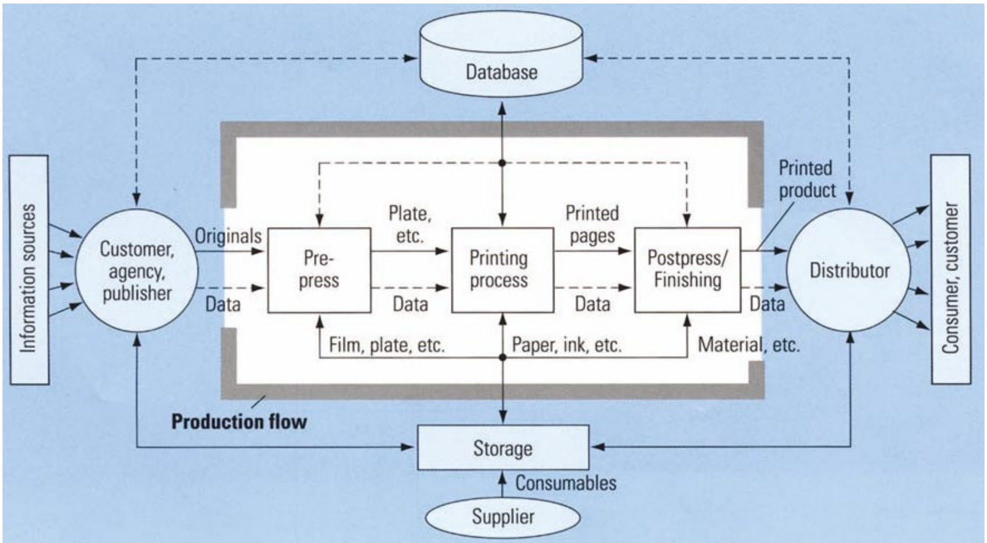


Figure 40: Production, material and data flows in the main phases of print media production ((Kipphan, 2001)).

[29] Courtesy of https://www.printingnews.com/print-finishing-mailing/postpress-finishing-equipment/product/10012709/standard-finishing-systems-standard-horizon-bq270c-singleclamp-perfect-binder, used under fair use policy.
[30] Courtesy of https://brainrack.co/recognise-ideal-6660-guillotine-easily/, used under fair use policy.

Modeling an end-to-end process for a print media order in BPMN results in Figure 41. The process is modelled in a layered approach, with every individual process model referring to a main activity. Order requirements are first checked whether the facilities are capable of producing it. Planning involves the determination of the production path. The model considers the example of a book to be printed, thus parallel printing activities are initiated, one for the cover, one for the bookblock. Merging the two printed semi-products requires some finishing activities and then the order is ready for warehousing and distribution.

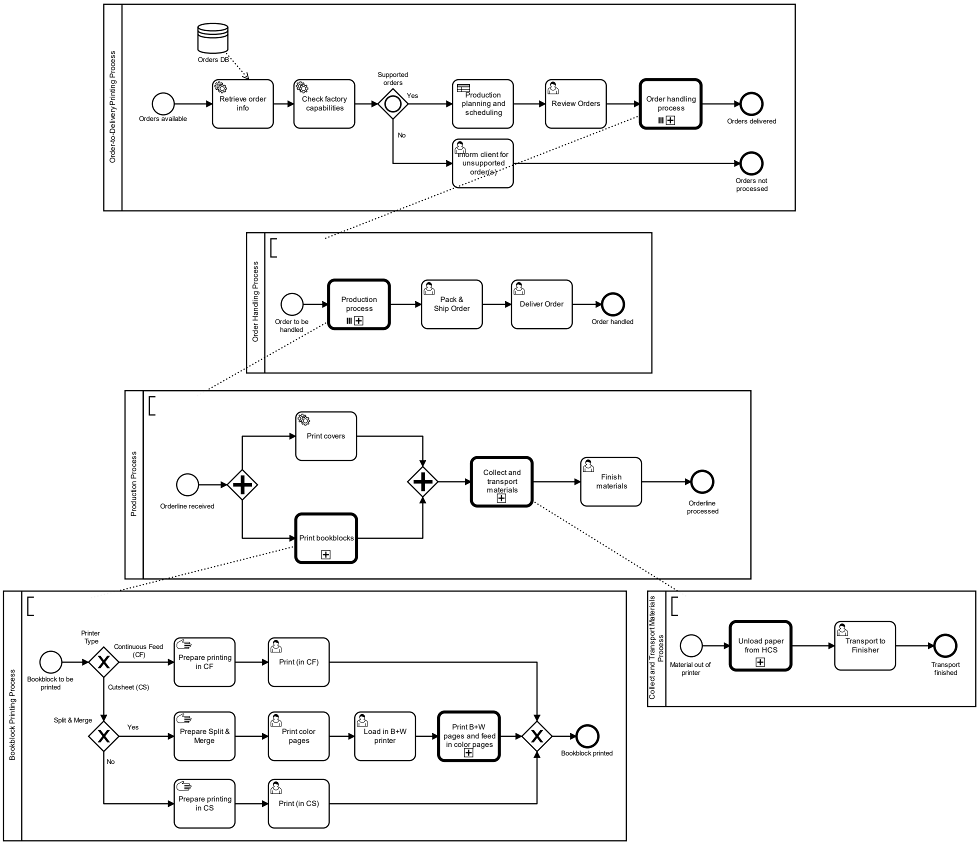


Figure 41: End-to-end process of a printing order, modelled in layered BPMN processes.

While the actual printing is performed in a fully automated way by the printers, manual activities are required by human operators. These include loading the right material to the right printer, unloading printed media into storage units, checking errors raised by printers (not modelled in Figure 41 for simplicity reasons), etc.

2.2.3.3 Support systems

The process automation among standard equipment encountered in a print shop has been enabled by the standardization of the exchanged data. The International Cooperation for the Integration of Processes in Prepress, Press and Postpress Organization (CIP4[31]), defined the

[31] https://www.cip4.org/

XML-based Job Definition Format (JDF) open standard[32]. JDF uses the term (print) job to describe the output that is desired by a customer. A job details both the resources/components (devices and their software) and the processes required to produce a final or intermediate product. The communication among the resources is done via the Job Messaging Format (JMF) protocol (also defined in the JDF standard), enabling their integration. With JDF, printing firms can also exchange production information with their enterprise information systems (EIS), such as Management Information Systems (MIS), Production Management System (PMS) or Manufacturing Planning and Control (MPC) system. These are typically used for administrative functions, order handling, resource planning, etc. That integration is made even easier with the newest XJDF (Exchange Job Definition Format) standard[33], which is a simplified version of JDF and is designed to be a pure information interchange interface. These standards though, do not actually execute the specified activities. Software applications that enact the JDF production paths are required.

Apart from using the JDF standards for defining workflows, other approaches have been developed, such as by using the Petri net notation (Gottumukkala & Sun, 2005). That approach provides both modeling and assessment, through simulation, of production processes, but no execution support in terms of coordination and allocation of tasks.

The benefits of the workflow automation are not scoped only to production, but are expanded to, and even beyond, pre- and post-production stages, where customers are also involved. Print on demand (PoD) concepts are realized with workflow and process management applications (Glykas, 2004; Zhu & Li, 2012). A workflow management system has also been introduced in Wang & Su (2011). Similarly, Wang et al. (2018) propose workflow technology, based on BPM, in printing and publishing industry.

2.2.3.4 Identified problems

Various issues have been identified in the current way of operations at various levels.

At operational level, the media loading and unloading onto/from are broadly performed by human operators. The recent years, examples of robots performing these tasks have already been introduced[34,35], but the issue is broader than the physical motion of loading/unloading. The routing of printing jobs and orders is a knowledge intensive process. At the input side of a printer, operators keep track of the job queues of the various printers, and the different types of media which are required to print these jobs. Based on the schedule, they create mental "shopping lists" in their mind of what media to collect from the media store and to what printer they should bring it, at what moment in time (preferably just in time for a print job to start). Similarly, output from multiple printers needs to be collected and brought together at a post-processing machine, in order to be combined in a finished application product. (e.g., bookblock and cover need to come together at a binder). Every application has its own

[32] https://www.cip4.org/files/cip4/documents/JDF%20Specification%201.7%20www.pdf
[33] https://www.cip4.org/files/cip4/documents/XJDF%20Specification%202.1.pdf
[34] https://static1.squarespace.com/static/576137c607eaa0ea7778f49b/t/5937d5ffb3db2b977427569f/1496831490450/Co-working+robots.pdf
[35] https://printbusiness.co.uk/news/Robots-are-coming-to-a-print-plant-near-you/106315

(unique) route but as it is not viable to transport each of them independently (to avoid excessive walking), transportation is often done in large batches. Because the different parts of a job are often not printed at exactly the same time, some parts need to be stored in a work-in-progress (WIP) place, waiting to be collected when the rest of the job has also been printed. Figure 42 illustrates the coordination of post-pressing transportation, currently performed by operators.

Obviously, this way of working causes physical and mental stress on operators. Consequently, more errors might appear (e.g., wrong media in a printer) or delays due to missing media on time, resulting in inefficient production.

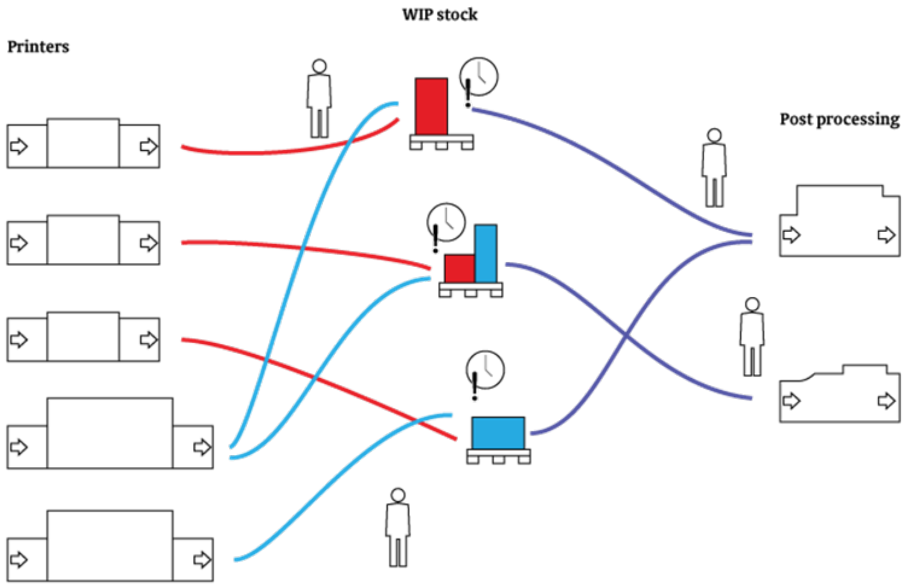


Figure 42: Current post-pressing flow, with human operators to coordinate the transportation of semi-finished products to work-in-progress (WIP) stock and finishing machines (source: OEDIPUS project's material).

At a physical level, handling media and products for long time is a tedious task. Forklifts are used but in many cases the activities are performed completely by hand, in non-ergonomic postures, as shown in Figure 43.

Figure 43: Unloading paper from printers, either completely by hand, or assisted with a forklift (source: OEDIPUS project's material).

At systems support level, while all the standards and workflow applications mentioned in Section 2.2.3.3 have enabled the production process automation, an end-to-end process-oriented coordination is often disregarded. Moreover, there are still activities, especially in the post-press stage – where the physical aspect is very relevant –, for which integration and orchestration of non-standard resources for printing environments (e.g., AGVs, robotic arms) is required and not yet matured. A process management system is missing to provide control among heterogeneous resources and a broad overview of operations.

2.2.3.5 Intervention scenario(s)

Within EIT OEDIPUS project, CPP has defined two intervention scenarios, towards realizing the factorization of printing. Both refer to the post-pressing operations and are individually discussed in the following two subsections.

2.2.3.5.1 Automated paper unloading by a fixed cobotic arm

The paper unloading from a printer, currently performed by operators, will be performed by a collaborative robotic (cobotic) arm with a special gripper to grasp the paper. Figure 44 shows the design of the special gripper.

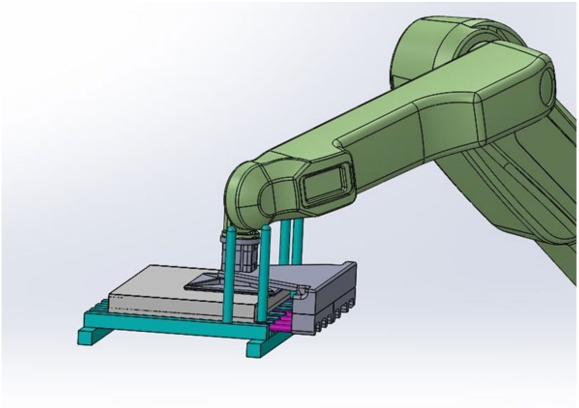


Figure 44: Design of a cobotic manipulator arm with a special designed gripper to unload paper from printers (source: OEDIPUS project's material).

The vision is that the cobotic arm will unload the paper from the printer and deposit it directly on an AGV, which in turn will navigate to the next post processing machine, as illustrated in Figure 45. With a sufficient number of AGVs, no WIP will be needed.

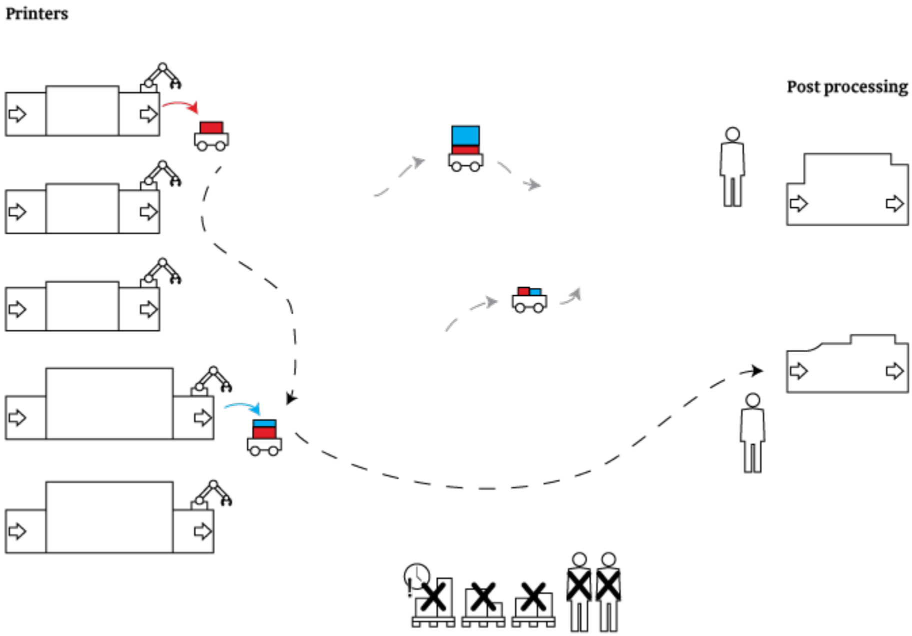


Figure 45: Future scenario for automated post-pressing transportation of semi-finished products, with fixed robotic arms to preform the unloading onto AGVs (source: OEDIPUS project's material).

The scenario requires orchestration of heterogenous actors by a process management system. That system shall be able to communicate with both the printer (to receive signals that paper has been printed) and the cobotic arm (to start the unloading).

For experimental and demonstrations purposes, the scope of this scenario is limited to one printer. A WIP place will be placed next to the printer, so the cobotic arm can deposit the paper in case no AGV is there yet. The scope is illustrated in Figure 46. Note that since the robotic arms is of collaborative technology, no safety zones are needed.

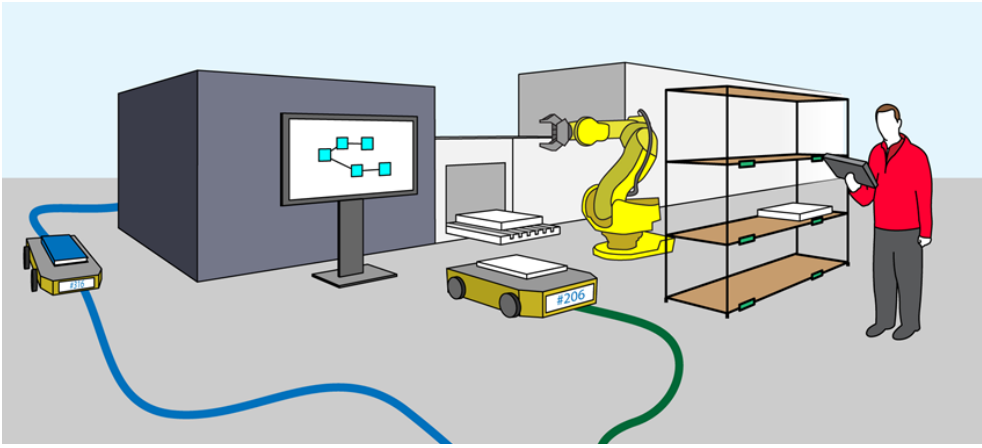


Figure 46: Illustration of the scope of the first intervention scenario (source: OEDIPUS project's material).

2.2.3.5.2 Automated collection of printer output by a mobile robot

The second intervention scenario extends the scope of the first by involving more printing equipment and replacing the fixed cobotic arm with a mobile robot. The mobile robot consists of an AGV with a cobotic arm mounted on it (with the same gripper as in the first scenario).

As more printers are included, AGVs can hop from one printer to the other, to collect output from the same printing job or to provide the requested material in the next device. A WIP place might be needed to cover excessive production volume. In that case, it should also be connected to the orchestration system so robotics know exactly on which shelves to deposit or pick parts. The vision is illustrated in Figure 47.

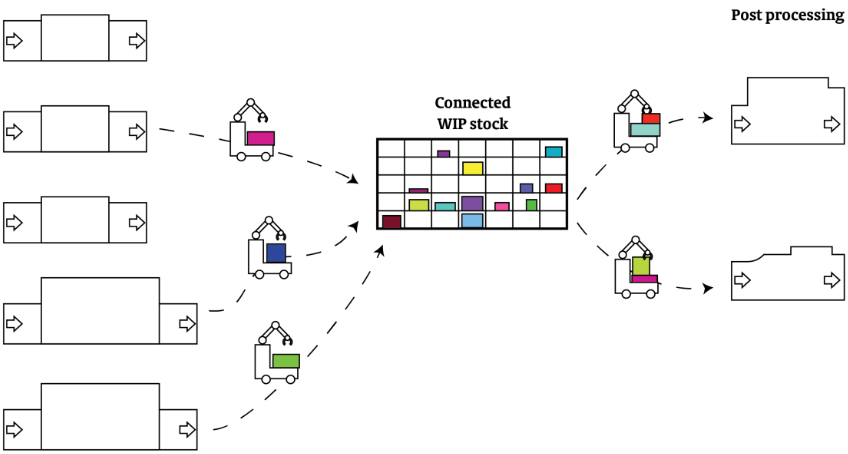


Figure 47: Future scenario for automated post-pressing transportation of semi-finished products, with mobile robots (AGVs with mounted robotic arms) (source: OEDIPUS project's material).

2.2.4 Robert Bosch España Fábrica Madrid (BOS)

Robert Bosch España Fábrica Madrid[36] (abbreviated as BOS in this thesis), a pilot partner in the SHOP4CF project, is the Automotive Electronics division factory in Madrid. In this location, peripheral acceleration and pressure sensors (PAS, PPS) for impact detection (e.g., on airbags) and ultrasonic parking assistance sensors (USS) are produced, shown in Figure 48, with approximately 64 million sensors manufactured yearly[37].

Figure 48: BOSCH automotive sensors: (a) peripheral pressure sensor (PPS) for side impact detection[38], (b) peripheral acceleration sensor (PAS) for impact detection[39], (c) ultrasonic sensor (USS) for parking assistance[40].

BOS has fully automated production lines for the automotive sensors. However, their utilization is not 24/7, but depends on the orders and the type of products that are required. As the utilization rate of these lines is random, it is difficult to schedule dedicated resources for performing activities. As a result, resources are asked sporadically to perform operations, while their main responsibility is other tasks. Moreover, some of these tasks are repetitive and not ergonomic (e.g., loading material onto machines). Thus, BOS aims at better resource utilization and reducing labour effort with the introduction of mobile robotics, in safe collaborative workspaces.

2.2.4.1 Production layout

Figure 49 structures the physical organogram of BOS, according to IEC62264-1 standard (see Figure 18). BOS participates in SHOP4CF project with two use cases, one addressing transportation and assembly issues at Printed Circuit Board (PCB) and Electronic Control

[36] https://www.grupo-bosch.es/en/our-company/bosch-in-spain/madrid-plant/
[37] Numbers in year 2011 according to https://www.grupo-bosch.es/en/our-company/bosch-in-spain/madrid-plant/
[38] https://www.bosch-mobility-solutions.com/en/solutions/sensors/peripheral-pressure-sensor/
[39] https://www.bosch-mobility-solutions.com/en/solutions/sensors/peripheral-acceleration-sensor/
[40] https://www.bosch-mobility-solutions.com/en/solutions/sensors/ultrasonic-sensor/

Unit (ECU) production areas, and one at Sensors production area and more specifically at PPS area. The latter use case is the focus of the current research.

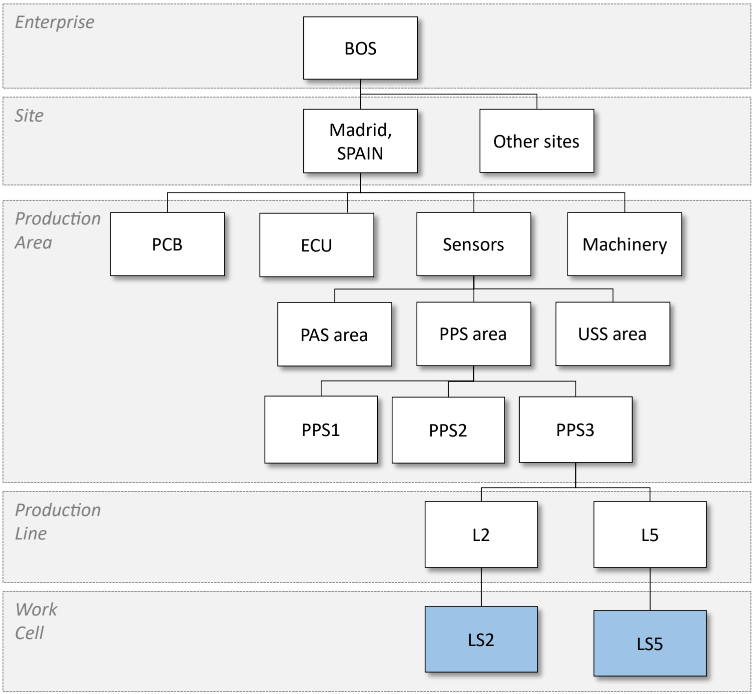


Figure 49: Physical hierarchy of BOS, according to IEC62264-1 standard, with the blue-highlighted work cells as the focus of the one of the two application scenarios within SHOP4CF project.

PPS3 involves production of the latest generation (3rd) of the peripheral pressure sensors (PPS). These sensors are installed in the door cavity, providing fast and robust side impact detection. The sensor continuously measures atmospheric pressure and detects any changes resulting from a deformation of the door. In comparison to central sensors, the peripheral pressure sensor makes it significantly easier to distinguish between an actual impact and harmless impulses. By means of continuous pressure distribution within the door interior, the entire door acts as a sensory element[41].

Two similar production lines (labelled as L2 and L5) in PPS3 are under concern of this research. Upon manufacturing of the sensors, they are automatically transferred on a conveyor belt to the last station of each line, which is a packaging station, as visualized in Figure 50.

[41] https://www.bosch-mobility-solutions.com/en/solutions/sensors/peripheral-pressure-sensor/

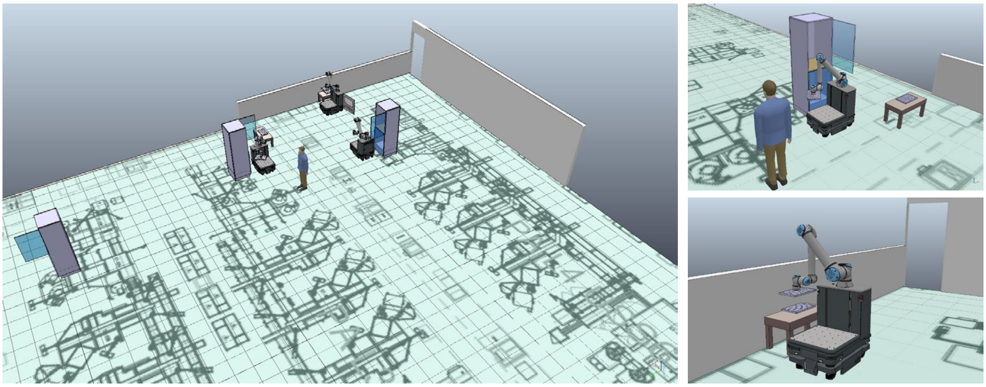


Figure 50: Digital representation of part of the production area, with two loading stations at two similar production lines[42].

The packaging task is performed by a fully automated industrial robot, which places the sensors into protective blisters (trays). The robot works in a closed, squared station. On the one side of the station, empty blisters are supplied onto palletizers. Two towers of maximum 12 blisters each can be placed onto the palletizers. The inner (i.e., closer to the robot) tower is automatically fed to the robot (one blister at a time). When the first tower is "consumed", the second tower is automatically shifted closed to the robot (so ready to be fed). A PLC sensor recognizes then that no second tower exist, so it raises a signal for more blisters. On the other side of the station, blisters are coming out one by one, filled with the sensors (the third side of the station is the conveyor belt that brings the produced sensors. The fourth side is closed).

As there produced several types of sensors, a number of different blister types are used, as the ones shown in Figure 51. The blisters are available in wagons, close to the packaging station (a material provider takes care to bring empty blisters in the wagons).

Figure 51: Types of blisters to package the produced sensors for damage-free transportation[43].

2.2.4.2 Manufacturing process

In the current situation, a human operator is responsible for providing the required blisters to the packaging station. He is also responsible to put the filled blisters into cartons, ready to be transported in warehousing. However, the operator is primarily involved in the other

[42] Courtesy of JVERNES – https://www.irt-jules-verne.fr/en/, used under fair use policy.
[43] Courtesy of BOS pilot, used under fair use policy.

activities in the PPS production area, thus the loading task (mainly) introduces disruption to his work. Moreover, as he might be busy in other tasks when the robot requires blisters, then the packaging task is paused, causing delays.

The current packaging process is described in BPMN, shown in Figure 52. Note that the arrow from the signal start event to the first task does not advance the process automatically. When the PLC system of a loading station raises a signal for empty tray on its palletizer, the task for the operator is not triggered automatically, as in the as-is situation there is no system to inform him. There is a light beacon to physically notify the operator, who in any case has to observe manually that loading is required. An independent process for provisioning empty trays into the wagons is modelled as well. This, also, is not an automated trigger as there is no sensor for identifying empty wagons, but instead the material provider has to take care of supplying in due time.

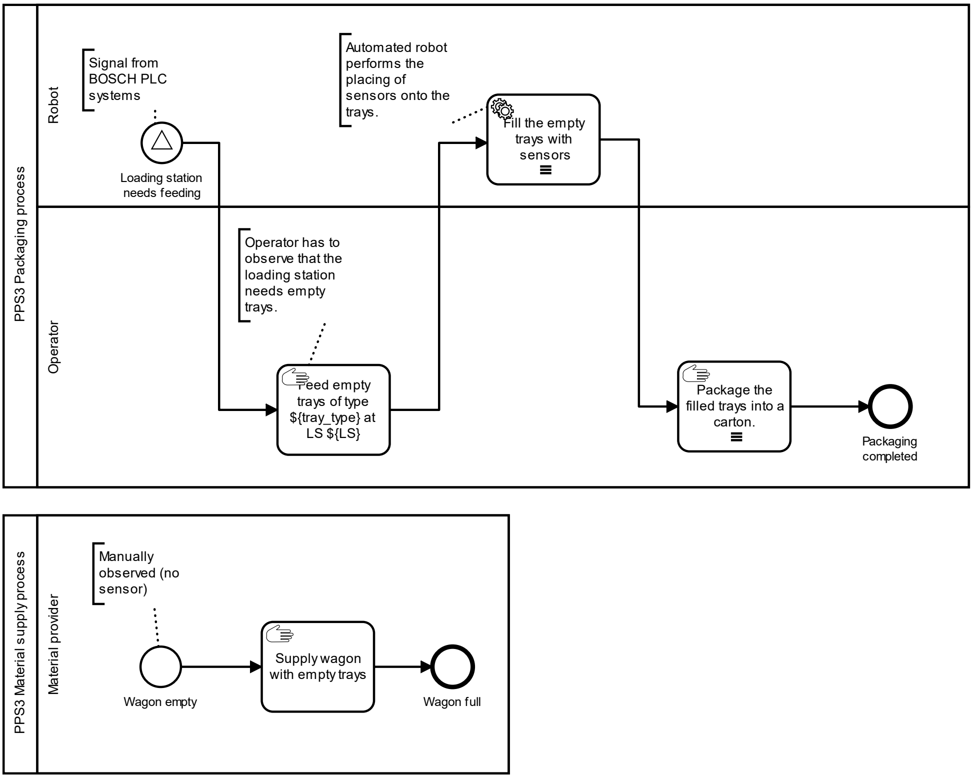


Figure 52: BOS current packaging process of PPS sensors, modelled in BPMN.

2.2.4.3 Support systems

PPS3 is a fully automated production area with a lot of industrial machinery and equipment, controlled by their respective PLC systems. An MES system takes care of the automation and scheduling. For privacy reasons, BOS could not share more information on the type of systems they use.

2.2.4.4 Identified problems

Due to the random production of sensors, resources are underutilized. While this might not be an issue regarding the packaging robot (as it halts its motion when there are no available sensors or blisters), it is problematic for human operators. They have several work cells under their responsibility and during their shift they walk around the production lines to ensure that they operate normally, taking corrective actions when needed. Loading empty blisters onto the packaging stations disrupts their main tasks. Furthermore, the task can cause fatigue as it is repetitive and not ergonomic (as they might have to bend to pick up empty blisters and load them onto the stations).

Delays on loading packaging stations with empty blisters cost time and money. Accurate planning is not feasible to avoid any delays, due to the random occurrence of activities. A fully automated solution is necessary. However, the strict cycle times have to be respected.

2.2.4.5 Intervention scenario(s)

The current packaging process of PPS sensors will be enhanced with the introduction of an AGV with a robotic arm mounted on it, to be shared across multiple lines. The mobile robot will be responsible to pick up empty trays from the wagons and load them onto the palletizer of the loading stations. It will serve both loading stations. The human operator will be called to load blisters onto a station only when the mobile robot is busy loading onto the other station. He can also assist in case the mobile robot raises an error (e.g., low battery).

The optimization in resource utilization upon introduction of the mobile robot requires a process management system to orchestrate the activities and allocate the right tasks to the right resource at the right time.

The solution will relieve operators from repetitive tasks and allowing them to focus on their main responsibilities on other work cells. Moreover, the resource optimization will increase productivity and reduce costs.

2.3 Design requirements

This section summarizes the problems and challenges in smart manufacturing, as identified both in literature and practice, with the purpose of eliciting requirements for the design of solutions. Upon design, realization, and demonstration of the solutions, the requirements are referenced again during evaluation (Chapter 7), to verify that the proposed solutions satisfy them.

The general problem of complexity in manufacturing operations in the current era is decomposed into three different perspectives. Demand chain models in "high mix-low volume" production environments, together with the increasing technology push for modern manufacturing, result in complex production scenarios (Section 2.1.6.1). For instance, the high customization of products and the variability in production routes at CPP (Section 2.2.3) pose difficulties on the automation of operations with the current practices. Dynamic markets, with high demand uncertainty, and introduction of promising, yet immature technologies have given rise to disturbances in normal production operations (Section 2.1.6.2). The case of TRI (Section 2.2.2), with high production and equipment variability to produce highly customized and small batches, is a representative example of a manufacturer with high risk of production downtimes due to equipment malfunctions. Heterogeneous technologies and fragmented robotic solutions result in integration complexity (Section 2.1.6.3). For example, at BOS case (Section 2.2.4), the need for a shared mobile robot across work cells requires higher level orchestration of different technologies. The integration that Industry 4.0 demands, is also hindered by disconnected enterprise information systems (Section 2.1.6.3), which has been encountered at TRI case.

The set of problems and challenges, considering also key domain characteristics, are the source of a list of high-level requirements that the designed solutions should respect. As the individual conceptual designs for tackling respective aspects of the process complexity issue (presented in Chapter 3, Chapter 4, and Chapter 5) are consolidated (see Figure 13) into an integrated system architecture (presented in Chapter 6), the requirements focus on the final artefact, i.e., to the advanced MPMS, and not to the individual conceptual designs. Table 5 lists the elicited requirements from both literature and practice. These are mainly grouped as design/execution requirements, respecting the two main phases of traditional BPM lifecycles (Brocke & Rosemann, 2010; Dumas et al., 2018; van der Aalst et al., 2003; Weske, 2012) and the distinction between type (design) and instance (execution) along the *life cycle & value stream* dimension of the RAMI 4.0 framework (DIN/DKE, 2016). The requirements are stated based on the “Easy Approach” syntax (Mavin et al., 2009; Mavin & Wilkinson, 2010):

<optional preconditions> <optional trigger> **the** <**system name**> **shall** <**system response**> …

This simple structure forces the separation of the conditions in which the requirement can be invoked (preconditions), the event that initiates the requirement (trigger) and the necessary system behavior (system response). Preconditions and trigger are optional, depending on the requirement type. The order of the clauses in this syntax is also significant, since it follows temporal logic:

- Any preconditions must be satisfied otherwise the requirement cannot ever be activated.
- The trigger must be true for the requirement to be “fired”, but only if the preconditions were already satisfied.
- The system is required to achieve the stated system response if and only if the preconditions and trigger are true.

The verification type presented in the table denotes how to check whether each requirement is satisfied upon design, development and realization of the solutions. According to (International Council on Systems Engineering, 2015), seven types of verification exist: inspection, analysis, demonstration, test, analogy, simulation and sampling. Quantifying the impact of the implemented solutions is rather difficult to be achieved, as there can be many factors that influence a system and its processes. Therefore, the developments presented in this thesis rely heavily on demonstration of solutions to *show correct operation of the system against operational and observable characteristics without using physical measurements*.

Table 5: High-level requirements for the design of an advanced MPMS as derived from literature and practice.

R#	Requirement	Literature source(s)	Practice source(s)*	Verification type	Verification scenario(s)
Design					
R01	The MPMS shall provide modeling support of complex processes that involve synchronization of activities by various actors (including human-robot collaboration scenarios)	Section 2.1.6.1	CPP TRI	Demonstration	CPP
R02	The MPMS shall be able to define manufacturing resources, such that it can determine during execution which resource should perform an activity (linked to R06).	(Erasmus, 2019)	BOS CPP TRI	Demonstration	(Erasmus, 2019)
R03	The MPMS shall be able to define tasks, such that clear control of activities is provided in both modeling and execution phases.	Section 2.1.6.1	BOS CPP TRI	Demonstration	BOS CPP TRI
R04	The MPMS shall be able to represent the physical equipment hierarchy, such that functional processes are mapped to their respective physical environment.	Section 2.1.2	BOS CPP TRI	System realization	BOS CPP TRI
Execution					
R05	The MPMS shall be able to enact the modeled processes in an automated way.	Section 2.1.3 Section 2.1.5	BOS CPP TRI	Demonstration	BOS CPP TRI
R06	The MPMS shall be able to dynamically select and allocate the most suitable resource(s) to tasks, based on task requirements and resource capabilities.	(Erasmus, 2019)	None	Test	(Erasmus, 2019)

R07	The MPMS shall be able to send a list of tasks to be performed by each actor in the production environment, for a specific production order.	Section 2.1.5.2 Section 2.1.6.3	BOS CPP TRI	Demonstration	BOS CPP TRI
R08	The MPMS shall be able to accept notifications from actors in the production environment regarding a change of manufacturing system status, including actors' availability and status.	Section 2.1.5.2 Section 2.1.6.3	BOS CPP TRI	Demonstration	BOS CPP TRI
R09	The MPMS shall be able to receive events regarding changes of the manufacturing system status.	Section 2.1.6.2	BOS CPP TRI	Demonstration	BOS CPP TRI
R10	The MPMS shall be able to react on exceptional events that change the status of the manufacturing system	Section 2.1.6.2	BOS CPP TRI	Demonstration	BOS TRI
R11	The MPMS shall be able to monitor the status of the manufacturing system during execution of processes.	Section 2.1.3 Section 2.1.7.4	BOS CPP TRI	Demonstration	BOS CPP TRI
General					
R12	The MPMS shall be able to provide administration of processes.	Section 2.1.7.4	None	System realization	BOS CPP TRI
R13	The MPMS shall be able to integrate to other EIS, including ERP/MES.	Section 2.1.5 Section 2.1.6.3	BOS CPP TRI	Demonstration	BOS CPP TRI

*Note that this column lists only the three selected pilot cases, presented in Section 2.2. Other cases from the three projects (Section 2.2.1) might enhance the importance of a requirement but are not mentioned here as they have not been presented. The author of this thesis has been personally involved in the requirement analysis of many of these cases (listed in Appendix B) and studied the others within the frames of the projects.

R03 and R06 have been covered by the work of Erasmus (2019), but added here for sake of completeness (considering the importance of resource allocation). R12 is not explicitly elicited from practical cases but is added as a typical system functionality of a BPMS. R12 and R13 are not grouped under design/execution categories as they refer to both in general.

2.4 Chapter conclusion

Manufacturing is shifting into a new paradigm of high technology innovation, digital transformation, dynamicity, and flexibility. Industry 4.0 developments promise increased productivity, higher resources efficiency, safer working environments and labor cost reduction. Yet, the path to those promises is full of complexities.

The current chapter studied concepts of the manufacturing domain and identified problems and challenges that enterprises face in their transition to smart manufacturing. Based on the hindering factors, a list of requirements is derived to drive the design of artefacts presented in Chapters 3, 4, 5, and 6. The developed solutions are demonstrated, in Chapter 7, on testbed application scenarios that have been described in the current chapter. The solutions are verified, in Chapter 7, with respect to the elicited requirements and evaluated on their utility.

CHAPTER 3

Flexible Process Modeling

Manufacturing enterprises perform a wide variety of operations, ranging from business processes to low-level machinery procedures, following the functional hierarchy of control (see Figure 7). Modeling the operations and processes is essential for various reasons. Process models are used to capture and analyze requirements, identify and prevent issues, conceptually represent (desired) behaviors of involved parties and systems, enable discussions between various stakeholders, manage information in a structured manner, and provide specifications for automated process execution (Aldin & de Cesare, 2011; Dumas et al., 2018; Ludewig, 2003; Mendling et al., 2008). Of course, process modeling should be performed at different abstraction levels, each with its own representation and level of details, and targeting different audience (Szelągowski, 2019). Different notations, languages and graphical representations exist to model processes related to each of the functional levels. For instance, modeling of processes on control level is widely represented with Grafchart (Årzén, 1996), a graphical language for state-based, sequential and procedural systems (Johnsson, 2008). It applies ideas of object-oriented-programming, and it is based on Petri Nets and Sequential Function Charts (SFC), a popular programming language for PLC, described in IEC 61131-3 standard (IEC, 2013a). Similarly, processes managed by an MES are often described with proprietary graphical languages (Gerber et al., 2014) or domain-specific modeling languages (Vještica et al., 2021). Business processes on enterprise control level of the hierarchy are traditionally modeled in BPMN, Petri Nets, UML (Unified Modeling Language) Activity Diagrams (AD) (Grady, Rumbaugh & Jacobson 1999) or EPC (Event-driven Process Chain) diagrams (Nüttgens et al., 1998).

The need for integration between business and manufacturing processes is essential in smart manufacturing, which asks for flexibility, transparency, and efficiency, as has been discussed in Chapter 2. Consequently, such a need has triggered interest to investigate how processes on various levels that have been modeled in different notations can interact to each other. In that respect, Gerber et al. (2014) examined the seamless integration between process modeling on business and production levels, by combining and transforming BPMN and Grafchart visualization languages. Conversely, Prades et al. (2013) proposed the use of BPMN for business to manufacturing integration, through seamless orchestration of information exchanges between ERP and MES systems.

As the current research advocates the application of BPM in smart manufacturing, it worth investigating how typical languages of the paradigm that are commonly used for business process modeling can be used to model manufacturing processes as well. BPMN, as the de-facto standard for business process modeling (Chinosi & Trombetta, 2012; Decker & Barros, 2008), is widely used for business processes (Wohed et al., 2006). Its interdisciplinary understandability (la Rosa et al., 2011; Witsch & Vogel-Heuser, 2012) and the expressiveness with respect to integration to execution (Ko et al., 2009), make it a promising candidate for use in the discrete manufacturing domain. However, as the language originated from business sectors where information processing is prevalent, compared to manufacturing where physical entities are included, domain-specific characteristics and challenges might require extensions to make it applicable.

Thus, the purpose of this chapter is to provide an answer on how to provide flexible modeling of complex production processes (RQ3) by designing modeling constructs with the use of BPMN.

3.1 Chapter outline

Design science research, that the current research generally follows, is applied to answer RQ1, as illustrated in Figure 53 (which zooms-in the aspect under concern of the three-aspects design and development of Figure 13). The practical relevance is ensured by the specific requirements identified from the analysis phase and has been summarized in Section 2.3. More specifically, there should be modeling support in three areas of interest: i) task delivery to heterogeneous agents, i.e., how work instructions are delivered to the actors (either human or automated operators) that perform activities, ii) human-robot collaboration, i.e., the interaction between a human operator and a robotic device/equipment on performing required operations, and iii) the activities synchronization, i.e., the points in time and space dimensions at which two or more activities have to be synched. On the other hand, the academic rigor is safeguarded by applying existing knowledge on process modeling of manufacturing operations, as identified from scientific literature. The design activities, i.e., the design of modeling constructs for complex production processes, yield in distinct artefacts that should satisfy the desired requirements. More specifically, the design produces: i) task delivery patterns, ii) human-robot collaboration (HRC) patterns, and iii) an activities synchronization mechanism. As the main focus is on supporting the execution of manufacturing processes, physical nature of the processes and the executability semantics are the leading perspectives for the design.

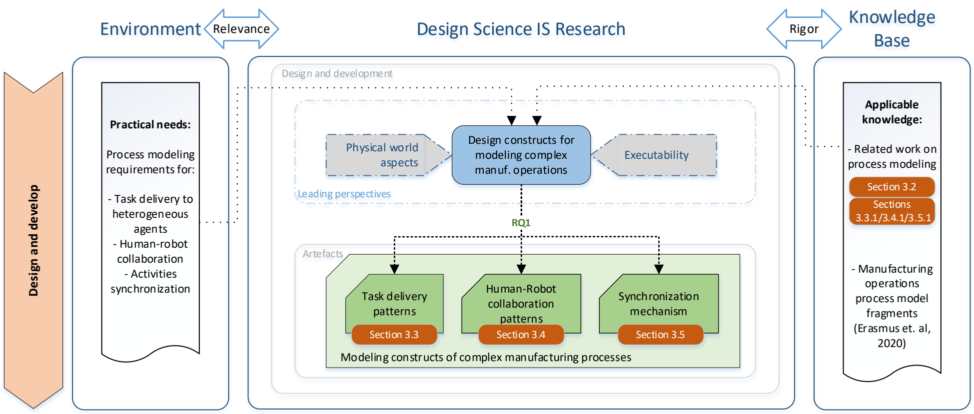


Figure 53: DSR approach for RQ1 - Flexible process modeling for complex production processes.

The fact that BPMN is the notation under consideration in the current research is a design choice based both on the increasing interest for its application in manufacturing (as discussed in Section 3.2) and the work of Erasmus, Vanderfeesten, Traganos, & Grefen (2020), which investigates the completeness and suitability of BPMN 2.0 for representing manufacturing operations. That work presents a set of process model fragments in BPMN 2.0 that can be used to represent the full taxonomy of manufacturing operations. Though, since its purpose is to use the notation for representing basic concepts of the manufacturing operations, further research steps are needed to make the process models fragments executable and consequently widely applicable.

This chapter is organized according to the DSR approach shown in Figure 53. Section 3.2 discusses related work on process modeling to explore what can be learned from similar efforts before the design activities have taken place. More detailed related work with respect to the designed artefacts are presented in the corresponding sections. Section 3.3 presents the design of the task delivery patterns. Section 3.4 presents the design of the human-robot collaboration method. Section 3.5 presents the activities synchronization mechanism. Section 3.6 concludes the chapter.

3.2 Related work on process modeling

Manufacturing domain, due to its physical nature, have inherent characteristics that need to be taken into account when modeling processes and production activities. For instance, a task "Move product from point A to B", requires information on physical location, weight of the object and even way how to lift it, and it might take considerable about of time which might be an important aspect towards efficiency. On the contrary, on business sectors many tasks typically process information instantaneously, transport information through digital channels, can easily copy, mutate or delete digital entities, which of course is not the case in the physical domain.

Thus, languages and notations for process modeling should consider the domain-specific characteristics and challenges. There have been many efforts to apply languages originated from administrative sectors into physical sectors, but also languages targeting especially the physical sector such as manufacturing. It is worthwhile then to study existing process modeling approaches to explore the current status, identify any gaps, and get inspirations. The exploration is performed on three areas of interest: ii) process modeling in manufacturing, iii) BPMN (as the notation under consideration) in (smart) manufacturing, and iii) general process modeling approaches, discussed correspondingly in the following three subsections.

3.2.1 Process modeling in manufacturing

Integration DEFinition (IDEF) methods refer to a family of modeling languages in the field of systems and software engineering[44]. Originally developed by the U.S. Air Force, it is heavily used in military and defense industries. The most common techniques for process modeling are IDEF0 and IDEF3. The former is used to model activities and decisions of a system or an organization in general. The latter is a capturing method to collect, describe and document processes. IDEF0/3 have been applied in manufacturing for modeling MES functions (Choi & Kim, 2010), supporting decision making in ETO strategies (Reid et al., 2018), process planning (Ciurana et al., 2008), or to design FMS (Pinarbaşi et al., 2013). Karaulova et al. (2019) use IDEF for modeling collaborative processes of humans and robots. Other applications exist in supply chain (Kuo et al., 2012) and construction industry (Kamara et al., 2000).

[44] https://www.idef.com/

Value stream mapping (VSM) is a lean manufacturing technique to model the flow of material and information in production environments (Rother et al., 2003) and has been widely used in manufacturing (Romero & Arce, 2017). It provides high-level views of processes and is mainly used to depict streams in serial production lines. That means that in more complex production scenarios the technique is not competent, and extensions are required (Braglia et al., 2006). Especially in the Industry 4.0 era where production environments are more dynamic, small batches and multi-variant production is prominent, and digital information is available from many sources, the concepts of dynamic VSM (Huang et al., 2019) and VSM 4.0 (Hartmann et al., 2018; Wang et al., 2021) have emerged.

3.2.2 BPMN in (smart) manufacturing

BPMN has already been used and extended in domains where physical entities are involved, such as healthcare and logistics. Braun et al. (2014) and Scheuerlein et al. (2012) have used and extended the notation for modeling clinical pathways. Though these applications do not "touch" the physical world. Khabbazi et al. (2013) propose the complementary use of BPMN and UML in production logistics, where UML covers the weakness of BPMN to model material flow.

The strengths of BPMN (as applied in various domains) made the notation a promising research area for application in manufacturing. García-Domínguez et al. (2012) compared BPMN 2.0 to IDEF3 and VSM. The study concluded that BPMN 2.0 can be seen as a superset of IDEF, with respect to schematic process representation, with the explicit addition of message exchanges, event handlers and process participants. Though, BPMN 2.0 is not able to model physical objects and their transitions. Compared to VSM, BPMN 2.0 is a complementary modeling approach, by adding more detailed process design specifications. As VSM does not provide any execution semantics, Zor et al. (2010) have proposed an approach to map VSM flows to BPMN models, which can be automatically executed. BPMN also compares favorably to other modeling languages such as Flowcharts, EPC diagrams, and UML AD. Araújo & Gonçalves (2016) find that BPMN 2.0 offers versatility that Flowcharts and EPC lack and considers the ability to describe complex processes as its main advantage. Entringer et al. (2021) argue that BPMN offers flexibility, while UML, EPC and IDEF cannot. Michalik et al. (2013) favor BPMN over UML for modeling the MES level functions, acknowledging though that the notation is not fully sufficient to capture the complexity of involved systems.

The limitations of BPMN have led researchers to propose various extensions for application in manufacturing. Zor et al. (2011) proposed extensions in aspects such as new activity types to cover manufacturing tasks, gateway elements to model material flow, and new resource types for machinery, parts and tools. However, these extensions refer to the representation semantiscs of the notation without execution support. Similarly, Sungur et al. (2013) extend BPMN for wireless sensor networks with new action types. Graja et al. (2016) have also proposed a BPMN 2.0 extension to handle CPS features, by introducing new task types and participants. No reference to execution is provided though. Yousfi et al. (2016) propose uBPMN (ubiquitous BPMN), which extends the BPMN meta-model with elements geared towards the smart factory including, but not limited to, sensory event definitions, data sensor representations and specific smart manufacturing task types. Petrasch & Hentschke (2016) propose the Industry 4.0 process modeling language (I4PML) which extends BPMN with various elements, such as sensing tasks, actuation tasks, IoT devices, human-computer interfaces and real/device data objects. Considering safety risks, Polderdijk et al. (2017) have proposed visualization extensions on BPMN process models.

3.2.3 General process modeling approaches

There exist many process modeling languages and approaches, following different paradigms and styles. There are imperative or declarative languages (Fahland et al., 2009; Pichler et al., 2011), those who focus on process specification through process fragments, or hybrid approaches. It worth exploring, then, what these offer for modeling manufacturing processes.

While BPMN is an imperative language, by (over-)specifying how to perform some work, in declaratives languages an outside-in approach is followed, meaning that any behavior is permitted unless explicitly restricted by some constraint(s). Declarative languages are encountered in Adaptive Case Management (ACM) (Motahari-Nezhad & Swenson, 2013) and Dynamic Case Management (DCM) (Swenson, 2010, 2013) paradigms, where goals are set but no explicit process paths are specified. The process participants are empowered to lead the execution and determine the achievement of the goal. Case Management Model and Notation[45](CMMN), (like BPMN, specified also by the Object Management Group[46] (OMG)), is a graphical notation for capturing work methods that are based on the handling of cases requiring various activities that may be performed in an unpredictable order. CMMN follows the guard-stage-milestone (GSM) approach (Hull et al., 2011) and structure cases in stages guarded by sentries, that denote conditions and events that should be satisfied in order a case enters or exits a stage. While the notation offers flexibility during execution, by letting knowledge-workers to select the next activity, it has received limited interest and uptake in industry.

ConDec (Pesic & van der Aalst, 2006) is another typical declarative language, with which constraints that must be adhered are depicted as relationships between the activities. DECLARE (Pesic et al., 2007) provides full support for loosely structured processes, with the use of constraints in process modeling. Similarly, Dynamic Condition Response (DCR) Graphs (Andaloussi et al., 2019; Hildebrandt et al., 2012) contain a set of events and five type of relations between them. Nested sub-graphs that can alternate completion states make DCR Graphs suitable in scenarios work has to be repeated. DCR Graphs were introduced to make run-time scheduling simpler and more intuitive for end-users, however the approach lacks concepts such as multi-instances, time and exceptions. In general, declarative languages offer flexibility in process execution and rely on the knowledge of the workers to advance a process. However, this type of modeling might not be suitable in manufacturing scenarios where such freedom is not allowed, procedures are more standardized, and automation does not depend on human knowledge.

In the broad scope of declarative approaches, process modeling with microservices offering (Stigler & Oberhauser, 2017) is a technique, borrowed from software development, for flexibility. Goal- and constraint-based agents navigate a dynamic landscape of semantically described microservices that form a dependency graph. Thus, workflows are dynamically constructed. The approach can be applicable in the physical manufacturing world, where human and automated agents can provide certain functionality through services which can be consumed by others. The concept of agents is also used by Fleischmann (2012), called

[45] https://www.omg.org/cmmn/
[46] https://www.omg.org/index.htm

subjects though, in the Subject Phase Matrix (S-PM) approach. The S-PM contains process activities, mapped to their subject (acting resource) and process phase. The matrix can be automatically converted into Subject Communication Diagrams (SCDs) and Subject Behavior Diagrams (SBDs), both of which are executable without additional coding required. S-PMs are a compact way to describe processes in an intuitive way, as processes are segmented into phases, as is often done in the manufacturing domain. The S-PM line of thinking is similar to the S-BPM.

The use of process blocks and fragments increase readability, understandability and maintainability (Gao & Jiang, 2009). Various approaches exist towards fragmented process modeling. Production Case Management (PCM) (Meyer et al., 2014) combines the modeling of small rigid process fragments with the flexible execution of ACM. A process model consists of process components that are combined during runtime in a stepwise goal refinement. The process components contain control flow nodes (similar to BPMN) and data nodes that specify pre- and post-conditions of activities. In PCM, knowledge workers are less needed (opposed to ACM), but there might be data synchronization issues or syntactical errors when components are combined, either by the modeler during design-time or the engine during run-time. Haarmann et al. (2015) introduce a prototype architecture and implementation of a PCM execution engine. Another approach of fragmenting process logic is the Task Based Process Management (TBPM) (Chung et al., 2003). The main component is a plan library containing a set of plans, each of which represented one possible way of achieving a task, by breaking it down into a structure of sub tasks. The plans are then linked together to from a process model. Through the use of domain ontologies as a means of shared vocabulary, the approach could be applied in manufacturing, though it mainly concerns the area of new product development.

In Aspect-Oriented (AO) business process management, a business process is separated into aspects (Jalali et al., 2013). An extension of BPMN, called Aspect Oriented Business Process Modeling Notation (AO4BPMN) has been introduced to model an aspect as a process fragment, containing one or more advices. Each advice captures a (part of a) concern under a certain condition. The produced models need to be interwoven for execution, either at design-time (static weaving) or during run-time (dynamic weaving). The approach reduces complexity and increases re-usability.

3.3 Modeling constructs for task assignment to heterogenous agents

According to van der Aalst & van Hee (2002), *a process indicates which tasks must be performed— and in what order—to successfully complete a case. In other words, all possible routes are mapped out. A process consists of tasks, conditions, and subprocesses*. The hierarchy of a process consisting of tasks is followed to model manufacturing processes. The concept of manufacturing task refers to the action(s) performed to complete a production goal (according to Luck (1995), an action is defined as a discrete event that changes the state of the environment). Activities, in general, are performed by actors (agents). In hybrid-actor manufacturing environments, i.e., where activities performed by both humans and machines/automated devices, often in collaboration, the notion of team has been constructed. As it is clearly described in the concept data model of the MPMS specification in Section 5.3, a manufacturing task is assigned to a team. Detailed execution of the task, described in steps, is performed by the members of the team. Tasks for which it is always clear which actor executes them, direct allocation is used, while for those tasks that a team of actors has to be compiled, an allocation mechanism is required. While for the former tasks a direct assignment can be easily modeled in BPMN, for the latter, a construct has to be created, which is embedded in a subprocess, as illustrated in Figure 54.

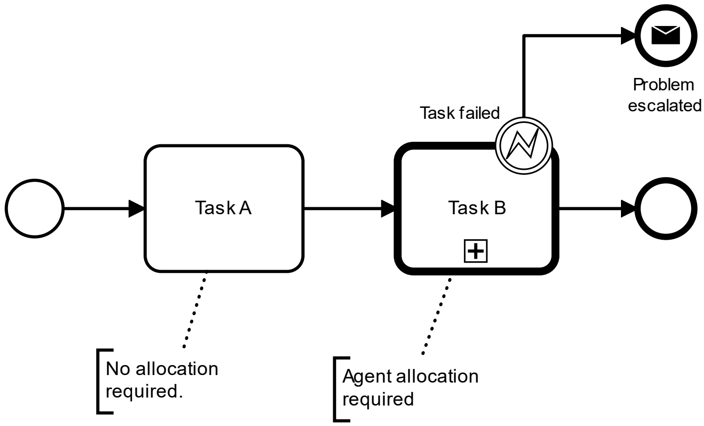


Figure 54: BPMN construct for tasks requiring no advance allocation (simple task) and tasks requiring allocation (subprocess, depicted with a bold border to highlight it).

An actor allocation mechanism determines the most suitable team to perform a task. Based on the outcome of the algorithm, i.e., team composition, a task is assigned to either a single human actor, or a single automated actor, or a team of actors. This is illustrated in Figure 55. The team decision algorithm is depicted as a business rule task, described in Decision Model and Notation[47] (DMN) (like BPMN and CMMN, specified also by the Object Management Group (OMG)). Details of the design and implementation of the decision algorithm can be found in Erasmus (2019). Note that for illustration purposes, tasks to be assigned to different type of actors are visualized with different task type icons, as have been designed in Aspridou (2017). Note, also, that the allocation mechanism includes basic exception handling functionality, which is not the focus of discussion here (elaborate details on exception handling is discussed in Chapter 4 and in Section 6.2.2).

[47] https://www.omg.org/dmn/

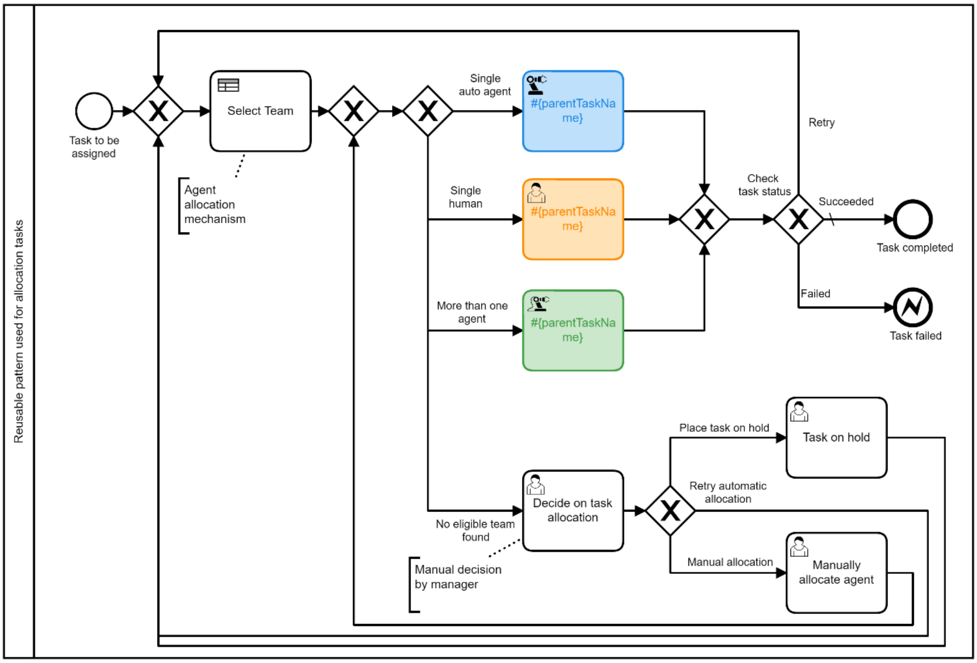


Figure 55: Task allocation mechanism to a team of agents (according to (Erasmus, 2019)).

Once it is determined which (team of) actor(s) will execute a task, it should be specified how this task is delivered. Section 3.3.1 discusses the task delivery patterns. Section 3.3.2 presents repetition patterns, i.e., when a task is assigned to the same actor for a given number of repetitions. Section 3.3.3 discusses the case of managing a queue of tasks to be assigned to an actor.

3.3.1 Task delivery patterns

The three colored-highlighted tasks of Figure 55, as the outcome of the allocation algorithm, require different delivery mechanisms. A typical BPMS provides tasklist applications for delivering tasks to humans, which have been modeled with the User Task type in BPMN. An example of a tasklist application is shown in Figure 56.

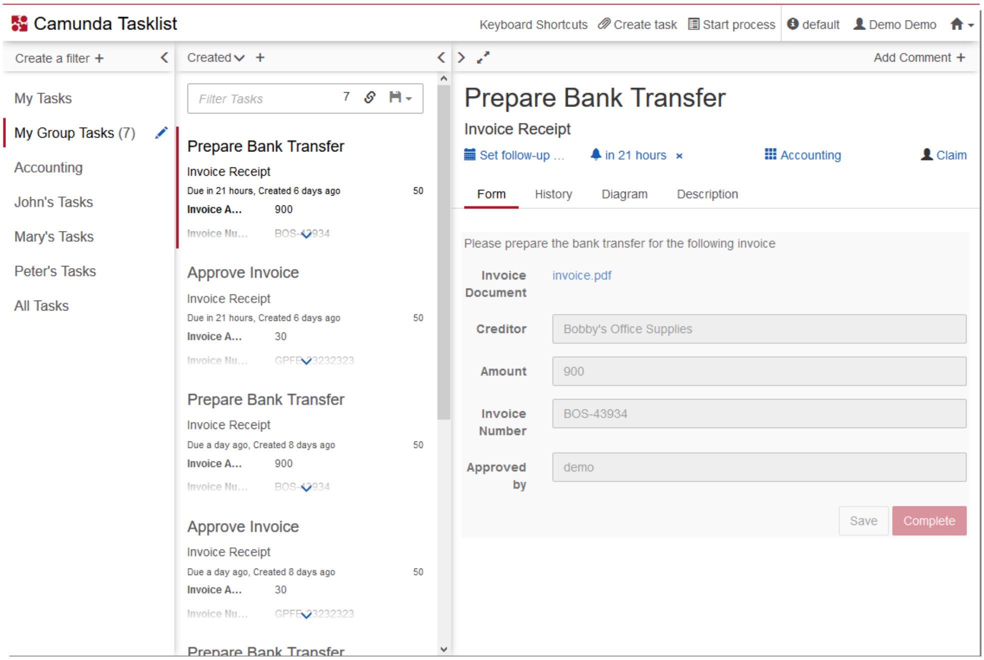


Figure 56: Example of Tasklist application for User Tasks (addressed to humans).

For tasks addressed to machines and robots, or to a team which includes at least one automated actor, tasklist applications are not appropriate. Therefore, different delivery mechanisms must be designed.

A main concept of a modelled task, from an execution perspective, is the waiting state it enters while it is being executed (worked on). A functional (and technical if feasible) split of a task (an activity in general) results in three main phases, as illustrated in Figure 57. In the starting phase, runtime actor allocation can be performed by specifying the actor(s) to which it is delivered (in contrast to fixed allocation performed during modeling). Considering the work item lifecycle of (Russell et al., 2006b), the states "offered" and "allocated" encountered in the Starting phase. During the Executing phase, the actual work is performed. The task can get in different states, e.g., started, in progress (elaborated in Chapter 6). In the Ending phase, the task is wrapped-up, by considering its state as completed or failed, and performing any actions that are triggered by the ending. For instance, a completion of a "Put product on shelf" task, can trigger the execution of a script to update the stock level. This often happens when small actions like in this example are not modelled explicitly (e.g., by a Script task following a User Task), but can be incorporated in the Ending phase of the main tasks that triggered them.

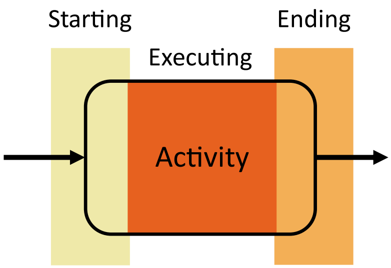


Figure 57: Main phases of a task in BPMN.

Considering that tasks for automated actors should be seen, from a functional perspective, as similar to the ones for human actors (establishing parity between the two types of actors), the aforementioned functionality has to be followed/preserved. Three different patterns are designed to achieve this, illustrated in Figure 58.

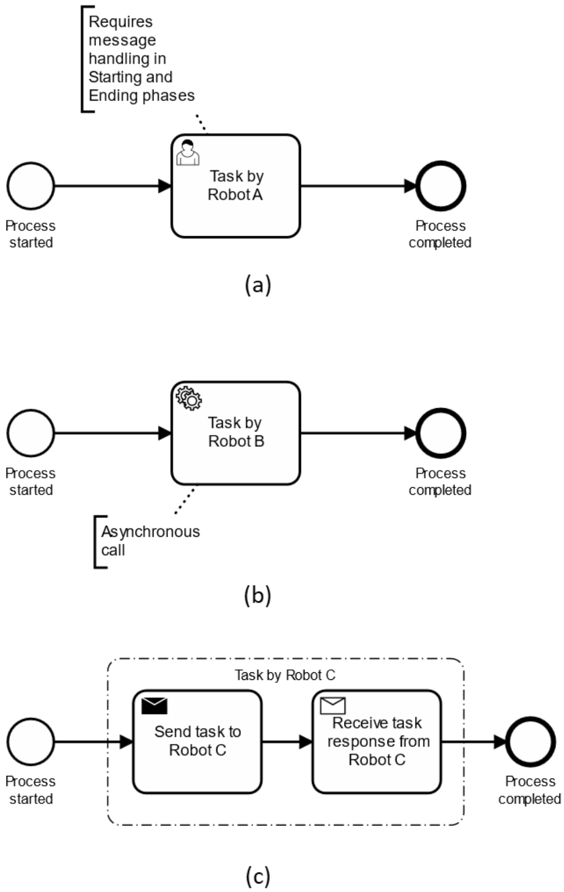


Figure 58: Task patterns for tasks addressed to automated actors (or teams that include at least one automated actor): (a) User Task, with embedded message handling in Starting and Ending phases, (b) Service Call in asynchronous mode, (c) Send and Receive Tasks.

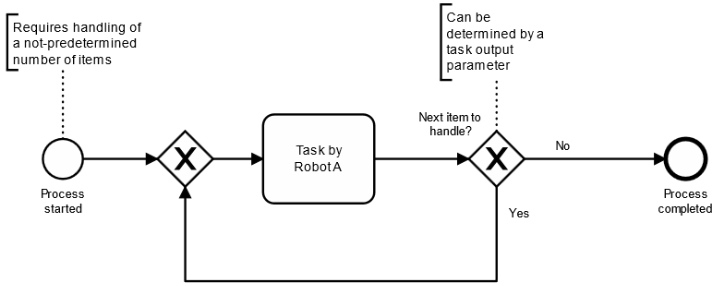
The first pattern (a) makes use of the User Task type offered by BPMN. This requires message handling. To avoid confusion with tasks addressed for humans, a different task type icon can be designed, like the robotic arm icon used in Figure 55. The second pattern (b) uses a Service Task but requires an asynchronous call to keep the task in a waiting state. The third pattern (c) is composed of the Send and Receive Tasks, which denote the sending and receiving task information (e.g., task input parameters – task completion status) in a more clear/distinct fashion, but both together preserving the desired waiting state. Note that corresponding Message Throw and Catch Events can be used instead of the Send and Receive Tasks. However, the task elements constitute a more intuitive way to represent the notion of a "task" to be delivered to an agent. Moreover, boundary events (e.g., error events) can be used on the Send and Receive Tasks, which might be useful in the scenarios.

While the above patterns are designed for automated actors, it is important to note that they can be applied for tasks addressed to humans as well. This is useful when (desktop) applications like the ones of Figure 56 are not possible or convenient for use (user-friendly) on a shop floor. Specially designed UIs are often required, managed by an external module (i.e., not part of a BPMS). Thus, when a process engine module of BPMS creates these tasks, the external module can receive and present them to the UI. The delivery to the external module can well be achieved with the above pattens.

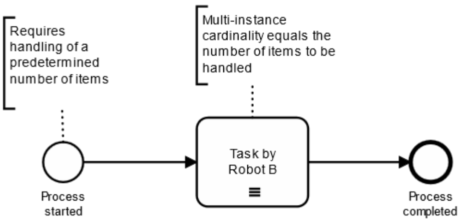
The operationalization of these patterns (e.g., message handling, service call) is discussed in Section 6.2.1.

3.3.2 Task repetition patterns

Often, in manufacturing processes, there are tasks that require some repetition. In case the repetitive work shall be performed by the same agent, thus, no re-allocation mechanism is required, the two patterns of Figure 59 are deigned.



(a)



(b)

Figure 59: Task patterns for iterative work: (a) non-predetermined number of repetitions, (b) predetermined number of iterations.

The first pattern (a) is used for cases where the number of iterations is not known prior to the initial assignment. A decision (XOR gateway) is taken upon each iteration is completed, whether the task should be repeated or not. The decision is based on the outcome of each iteration. For instance, filling a box with products where the remaining space cannot be determined in advance, but only when the box is full. The second pattern (b) is used when the number of iterations is known/determined a priori. A sequential multi-instance task is used, with the cardinality to be equal with the number of iterations.

These patterns are useful when the outcome of each iteration is relevant/interesting at the level of a BPMN process model. In case only a final outcome is required, the task can be modelled as a single activity and the iterations can be handled by a component/controller which handles the automated agent (defining as task input parameter the number of iterations, if known). Considering the example above, a task "Fill in a box with X products" can be modelled as a single task (no loops, no multi-instances), in case only the moment that the box is full is interesting at the BPMN process model level. Such separation is discussed in Section 5.3. Briefly to mention here, it depends on the global/local level of control that process management system shall have compared to a local orchestrator component (see Figure 93).

3.3.3 Task queue management construct

With respect to the level of control that a process management system shall have on the granularity level of a task, it is often required by such a system to handle a task queue. This is required when a local orchestrator component or a robot controller cannot handle consecutive assignment of tasks if a previous task has not been completed. Of course, if task queue management at BPMN process model level is not needed, then multiple assignment of tasks to a robot can be modeled as either a single task (and then the created task instances will be assigned) or a parallel multi-instance task.

In case task queue management is needed at the BPMN process model level, the pattern of Figure 60 is designed. The pattern shows an example of an agent able to perform two different types of tasks (type A, type B) (note that the tasks have been modeled as subprocesses, as they might need to apply allocation mechanism, as shown in Figure 54). When the agent is busy performing a task, a next request for a next task (either of the same or different type), received through the non-interrupting event subprocess, is registered in a queue. When the current (series of) task(s) is finished, the process returns back to check for available next tasks. Depending on the implemented queue policy (e.g., FIFO, LIFO), the next task is selected and assigned to the agent. The task is removed from the queue once it is completed (deferred removal), to avoid new requests to be assigned when the agent is still busy.

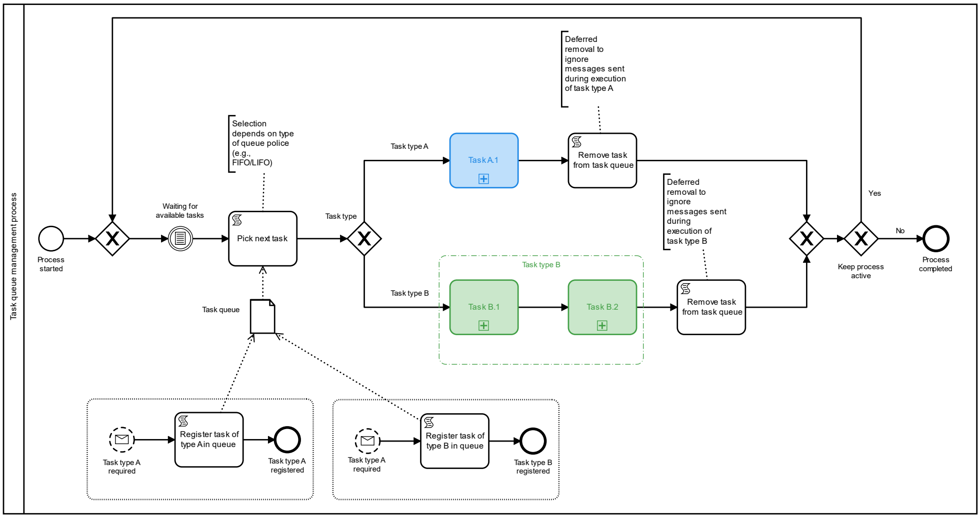


Figure 60: Task queue management pattern at BPMN process model level.

The reason why task type B is shown in the example is because it consists of a series of tasks which should be seen as one task entity. Let us consider the example of an AGV with a robotic arm mounted on it for performing some grasping actions, composing a mobile robot. If the task management queue refers to tasks addressed to AGV, an example of task type B could be asking the mobile robot to fetch an item from a shelf. In that case, the task B is split into a task B.1 for the AGV to move to the location where the shelf is and task B.2 for the robotic arm to grasp the item. In this scenario, when the robotic arm is performing the grasping part, the AGV should be still considered as busy, thus, unavailable to get a new task.

The pattern can be extended by introducing more task types, with corresponding non-interrupting event subprocesses, or by grouping together tasks (i.e., shifting the deferred removal of the task from the queue accordingly). Of course, the number of the non-interrupting event subprocesses can be minimized into one when the registration per task type can be achieved with a single script/function and no differentiation is required. For clear illustration purposes, one subprocess per task type is added in the example.

3.4 Modeling constructs for human-robot collaboration (HRC)

Human-robot collaboration (HRC) receives increasing interest in Industry 4.0 era, especially in "high mix-low" volume environments. Collaborative robots are employed to assist humans in assembly tasks (Karaulova et al., 2019; Miqueo et al., 2020). While adoption barriers have still to be addressed (Villani et al., 2018), with safety a major issue (Aaltonen & Salmi, 2019), HRC worth support from a process modeling perspective.

Section 3.4.1 discusses related work on using BPMN for modeling collaborative processes. Section 3.4.2 presents an approach to model collaborative process for assembly tasks. Section 3.4.3 presents modeling constructs to support deferred execution or parallel tasks.

3.4.1 Related work on BPMN for collaborative process

In the general interest on using BPMN for manufacturing processes (as discussed in Section 3.2.2), the notation has already been used to model collaborative processes. Schonberger et al. (2018) use BPMN in a new approach, called Human Robot Time and Motion (HRTM) for modeling collaborative tasks. HRTM combines the Methods Time Measurement (MTM) approach, used for modeling of working steps of human workers, and Robot Time and Motion (RTM) approach, used for modeling of working steps of robots. Froschauer & Lindorfer (2019) extend HRTM by combining it with ADEPT (Lindorfer et al., 2018), a universal modeling approach that allows a shift of programming complexity from the end user to a modeling expert. Though, the modeled BPMN workflows do not include an automatic trigger of robots to perform the corresponding command at the right time. Moreover, in a recent work by Schmidbauer et al. (2021) a digital worker assistance system is presented, based on the BPMN. The notation is used to enable non-professionals to create adaptive task sharing processes between human workers and cobots. Engels et al. (2018) extend BPMN in their language-based Adapt Case 4 BPM (AC4BPM) approach to model, among other IIoT processes, assembly tasks. However, execution support is only provided in an experimental environment. Finally, Knoch et al. (2018, 2020) use BPMN to model assembly tasks supported by AR, but no collaborative robot involved.

3.4.2 Modeling collaborative assembly processes

In collaborative assembly processes a robot and an operator receive instructions on performing similar tasks, but obviously executed differently. A similar approach to the one of Froschauer & Lindorfer (2019) is adopted to model such processes. However, the method presented here has two fundamental differences: i) A single BPMN swim pool is used instead of two swim pools. The single pool, with two swim lanes, one for the robot, one for the operator, eases the execution of the process by avoiding the extra communication messages between two pools. Moreover, in their approach it is not clear how a "Communicate" task is linked to a gateway element (on their model of Fig.1); ii) AND-split and merge gateways are used to achieve synchronization as tasks, without the use of time delays as in Froschauer & Lindorfer (2019). Introducing time delays for an actor until the other actor finishes his tasks (especially without any deferred constructs as described in Section 3.4.3) requires intelligence and precision of knowing the exact task duration values. Calculating these values might be complicated or even pointless to perform.

Consequently, a collaborative assembly process model is simplified, as the example shown in Figure 61. Modeling of collaborative processes in such a way supports also safety risk analysis (Pantano et al., 2022).

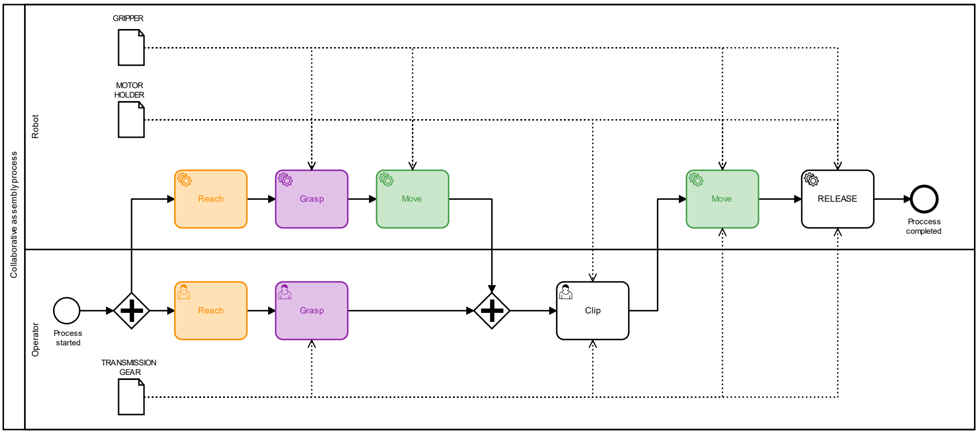


Figure 61: Process model for collaborative assembly processes.

3.4.3 Deferred task parallelism constructs

Task parallelism is a common construct encountered in collaborative processes, where activities are first distributed to various actors and then can be merged. However, there can be scenarios where a task shall be assigned once its parallel branch has started its actual execution. This scenario is called here as deferred task parallelism, i.e., two parallel tasks (or branches in general) in which the one has to be assigned when the other has started. Figure 62 illustrates the deferred task parallelism scenario (for simplicity reasons, the state "assigned" is used instead of the "offered" and "allocated" stated from the work item lifecycle of Russell et al. (2006b), as the relevant state is the "started").

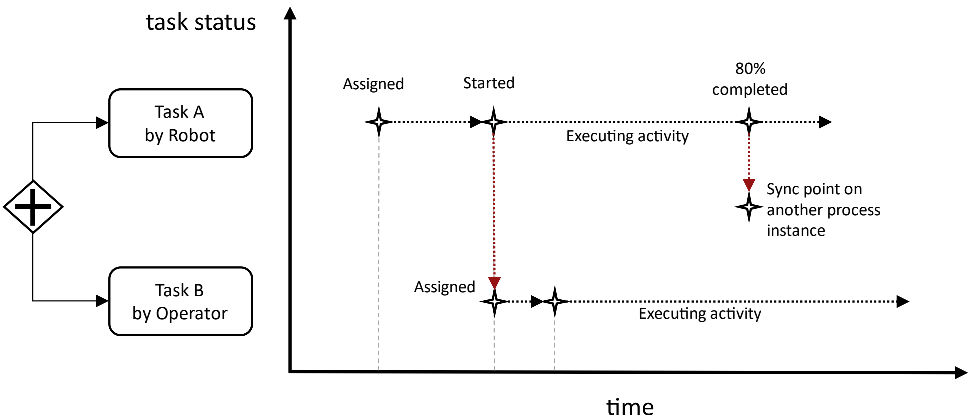


Figure 62: Explanation of deferred execution of parallel tasks.

Two modeling constructs are designed to support the deferred task parallelism. The first, shown in Figure 63, makes use of the non-interrupting boundary conditional event. Of course, the information of the actual starting point of execution of a task is sent by an actor (e.g., a robot through its controller software) and has to be caught by the right task instance. Especially in the case of many running instances of the same task definition, a correlation mechanism should be provided to trigger the right conditional event.

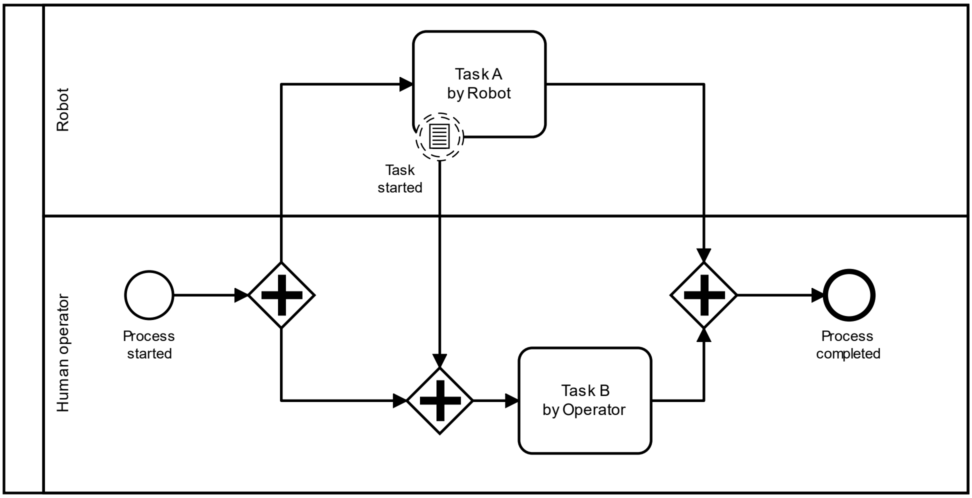


Figure 63: Modeling construct for supporting deferred execution of parallel task with non-interrupting boundary conditional event.

The second construct, shown in Figure 64, uses an intermediate conditional event and a non-interrupting event subprocess. The latter catches the information that task A has started and updates accordingly the condition variable of the conditional event. As both the conditional event and the subprocess are in the same process model (and thus in the same instance during execution), the correlation mechanism is easier compared to the first construct.

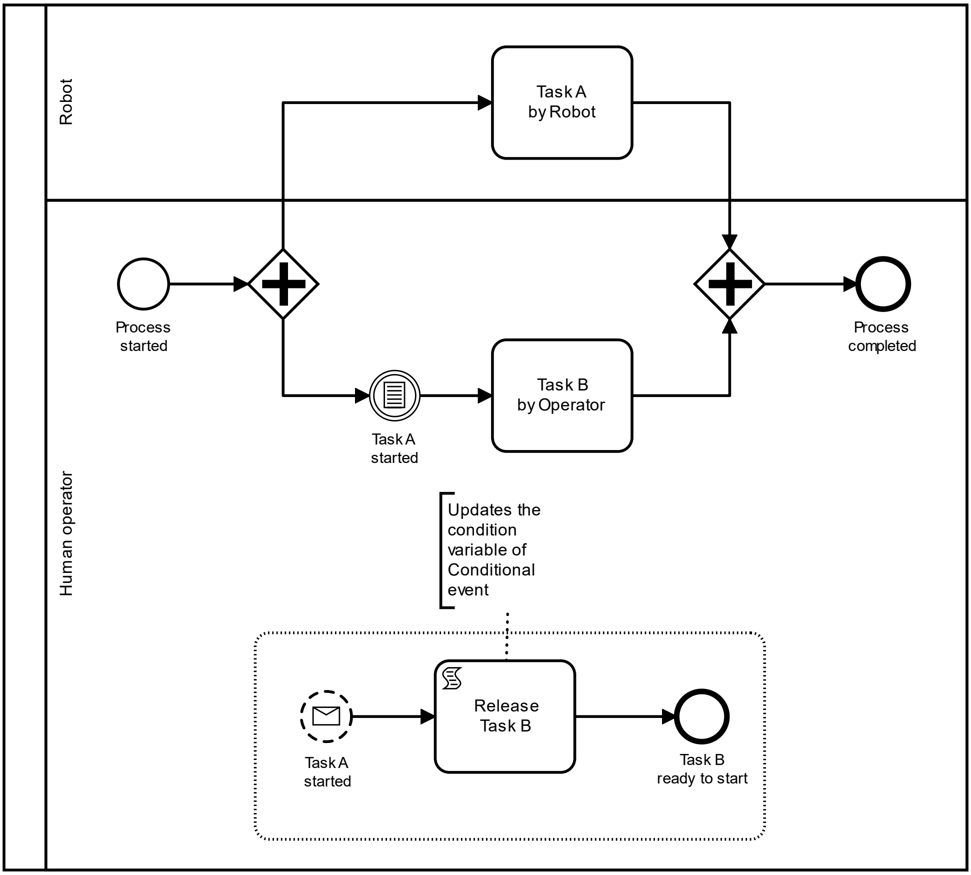


Figure 64: Modeling construct for supporting deferred execution of parallel task with non-interrupting event subprocess.

Deferred task parallelism can be also applied in any processes in which parallel activities of different actors have to be synced with respect to their enabling moments. In other words, the construct is not only applicable when two actors work collaboratively on the same task, but also when two different actors perform different type of activities. For instance, consider a task of a mobile robotic arm moving to a grasping position, while in parallel an item is conveyed to a picking point, but the grasping should start only once the item is at the right place. This is useful for activities optimization to reduce tact times. The construct can be also applied on any relevant events happening during task execution, not only with respect to their enabling moment (as illustrated in Figure 62 with the “80% completed” event of Task A). In case the synchronization refers to activities present in different process model definitions, or different instances of the same process definitions, a more advanced synchronization mechanism is required, as elaborated in Section 3.5.

3.5 Synchronization points

BPMN has already been applied and extended for modeling manufacturing processes, as discussed in Section 3.2.2. However, despite its maturity and recent interest, the notation has inherent limitations. One of these is the fact that process models in BPMN are designed from a single, isolated process instance perspective, disregarding possible interactions among instances during execution (Leitner et al., 2012; van der Aalst et al., 2017) . Often, process instances need to interact and collaborate based on information that is outside of the scope of one single instance. This collaboration is more important in manufacturing processes, where physical objects, and not only data information, are under consideration. Think of example of buffering points, where inventory is kept at an intermediate stage of a process, or the situation of bundling or batching products (multiple entities) for further processing as single entity (e.g., placing a number of items in a box for transporting). There should be hence, synchronization points where a process instance, representing the flow of activities of entities, waits or sends information regarding the state from or to other instances, commonly from different process definitions. BPMN provides basic synchronization with elements such as Signals or Messages. But the former is a broadcast message without any payload while the latter sends a payload message (with e.g., process instance identifiers or process definition keys) to only one instance. There is a lack of dynamic synchronization expressibility and functionality in the sense that the synchronization of the control flow of process instances cannot currently be decided based upon runtime state and content information of other process instances.

Buffering of entities and (un)bundling of entities and activities are constructs frequently encountered in the physical world of manufacturing processes. Using BPMN for manufacturing processes, entails explicit support for these constructs. This section designs such support.

The content of the current section has been published in Traganos, Spijkers, et al. (2020). Section 3.5.1 discusses related work on the synchronization shortcoming of BPMN. Section 3.5.2 analyzes the manufacturing constructs that require modeling support. Section 3.5.3 presents the design of a synchronization mechanism, called recipe system. The mechanism is operationalized in Section 6.2.1.

3.5.1 Related work on BPMN on synchronization points

The shortcoming of the language to support synchronization points has already been studied, but rather as a general problem, not targeting at the physical and manufacturing world. In general, we see two different paradigms; activity-centric ones (e.g., what BPMN follows) focusing on describing the ordering of activities, and artifact-centric ones focusing on describing the objects that are manipulated by activities (Cohn & Hull, 2009; Lohmann & Wolf, 2010; Meyer et al., 2013, 2015; Meyer & Weske, 2014). From a BPMN perspective, artifact-centric modeling support is limited, though extension elements to support the artifact-centric paradigm have been defined (Lohmann & Nyolt, 2011). Fahland (2019) approaches the process synchronization from a dualistic point of view, both from the activity-centric and the artifact-centric paradigm perspectives. The study argues that processes are active elements that have actors (agents) that execute activities. These actors drive the processes forward. Artifacts, on the other hands, are passive elements that are object to the activities. The activities are performed on these objects. While Petri nets are used as a means of process specification, Fahland argues that locality of transitions, which synchronize by "passing" tokens, are at the core of industrial process modeling languages, just like BPMN. Steinau et al. (2018) also consider many-to-many process interactions in their study, proposing a relational process structure, realizing many-to-many relationship support in run-time and design-time. Earlier work on process inter-actions by van der Aalst et al. (2001) (e.g., proclets), allowed for undesired behavior in many-to-many relations (Fahland et al., 2011).

Pufahl & Weske (2019) put forward the notion of a “batch activity”, which is an activity that is batched over multiple process instances of the same process definition. The batch is activated upon the triggering of an activation rule. The concept is similar to the approach presented in this thesis, but the current research includes a strong focus on the correlation of process instances of different process definitions, that typically contain different activities. Finally, Marengo et al. (2018) study the interplay of process instances and propose a formal language, inspired by DECLARE (Pesic et al., 2007), for process modeling in the construction domain.

3.5.2 Manufacturing constructs

Erasmus et al. (2020) have created a catalog of process fragments to model manufacturing processes in BPMN. The catalog is a result of rigorous research through a taxonomy of manufacturing operations translated into flow elements for representation in BPMN. The catalog has been evaluated for its completeness and suitability, and thus, is a solid and valid starting point for providing modeling support on specific manufacturing constructs.

The catalog includes process fragments for the four main categories of operations, namely production, quality, inventory and maintenance (Section 2.1.3). From the complete set, the ones that require synchronization of activities are under concern here. These are the ones referring to:

- Production operations
  - Material removal
  - Separating
  - Permanent joining
  - Mechanical fastening
- Inventory operations
  - Individual packaging
  - Unitizing
  - Buffering
  - Preservation
- Maintenance operations
  - Replacing
  - Scheduled replacing

The corresponding BPMN process model fragments are illustrated in Appendix C. As the operations in this set have similar characteristics, they are grouped into two main categories, namely, buffering and bundling/unbundling constructs (grouping shown in Appendix C). They are elaborated in Section 3.5.2.1 and Section 3.5.2.2 respectively, while Section 3.5.2.3 discusses the inherent limitation of BPMN to provide execution support for these constructs.

3.5.2.1 Buffering

From an operations management perspective, buffering is considered as maintaining excess resources to cover variation or fluctuation in supply or demand (Nahmias & Lennon Olsen, 2015). The concept is also referred to as decoupling inventory between process steps, as these can be performed independently from each other (Cachon & Terwiesch, 2009). From Defense Acquisition University DAU (2021), buffering can be defined as *a form of (temporary) storage with the intention to synchronize flow material between work centers or production* *steps that may have unequal throughput*. From the five types of inventories from de Groote (1989), the focus of the current research in on the decoupling inventory/buffers.

In the BPM field, van der Aalst (1994) had already argued that places in Petri nets correlate to physical storage locations, in his effort to use high-level Petri nets to describe business processes. Thus, from a process management perspective the notion of a buffer can be explained as follows. An instance enters the buffer and is kept in a holding state. Once a condition is met (e.g., capacity becomes available in the downstream production step), one or more entities are released. The selection of which entity to be released can be based on multiple queuing policies, e.g., the First-In-First-Out (FIFO) policy. Once an entity is released, control flow continues as normal.

The above explanation though, considers the buffering from a single process instance perspective, leading to the process instance isolation issue described above. There is a need to approach the construct from a process control perspective, as such that buffer-level attributes and information from many process instances are captured and managed, as illustrated in Figure 65.

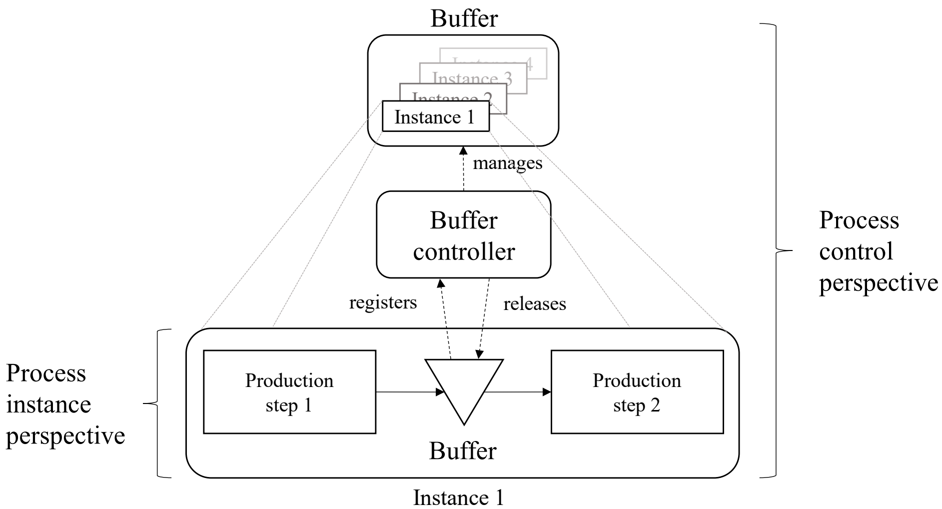


Figure 65: Buffering construct from both process control and process instance perspective.

3.5.2.2 Bundling and unbundling

Manufacturing operations literature recognizes operations that bundle, merge, unitize and package entities, as well as their inverse counterparts, but to the best of the researcher's knowledge no literature exists that describes how these entities are selected during operations. This is assumed to be described by the modelers in another part of the models or in different models. For the purposes of the current research, bundling is defined as *the synchronization of instances that are grouped in some way, either physically or virtually, whose control flow shall continue or terminate simultaneously as a group*. Note this is a process-oriented definition and caution should be taken for generalization.

Examples of bundling are commonly encountered when physical entities need to be grouped into some sorts of a container. Imagine for instance products being produced and put in a packaging box. Once the capacity of the box is reached, the box can be transported as a single entity. Upon arrival of the box to a distribution center, entities are unbundled again. The term bundling is used, as a more generic term instead of batching, since the latter normally refers to putting together entities of the same type, while in bundling entities of different types can be merged. Bundling is often encountered together with buffering, as quite often, (sub-) entities are buffered before the bundling operation can take place, to ensure all (sub-)entities are present.

3.5.2.3 BPMN support limitations on manufacturing constructs

A buffering point between two activities (or process fragments), as shown with the triangle element in Process Instance 1 in Figure 65, could be naively modeled in BPMN 2.0 with the use of conditional or (intermediate) message catching events. These elements can offer the "holding" state of the control flow. However, none approach is suitable. Conditional events use local-instance variables, ignoring information of other process instances. Message events are targeted to a specific, pre-defined in-stance, missing dynamic correlation information.

Bundling and unbundling constructs can be probably modeled with AND-gateways. But these gateways (un)merge control flows that can be modelled on the same definition, which is not always possible. In many scenarios, different processes have to be correlated and gateways cannot perform this. Multi-instance activities can be also used for "unitizing" entities (Erasmus et al., 2020). The spawning of repeated instances can serve (un)bundling functionality. However, the isolation problem appears here as well. Each child process instance is unaware of the information of the rest child instances.

3.5.3 Concept and functionality of a Recipe system

This section presents the design of a synchronization mechanism, called recipe system, to address the dynamic synchronization issue described above. The approach uses standard BPMN 2.0 elements to form a dynamic controller that works as a correlation mechanism for synchronization points amongst independent process instances. Section 3.5.3.1 discusses relevant concepts of the recipe system. Section 3.5.3.2 formalizes the description of these concepts.

3.5.3.1 Recipe concepts

The central notion of the system is the recipe. It corresponds to a synchronization (or integration) point, where (previously uncorrelated) control flows in independent process instances may be synchronized. It consists of a set of input rules and output rules. A recipe is fulfilled once all input rules are satisfied. Two important concepts are linked in a recipe. The instance type and the selector attribute. The first is used to group process instances of the same type in a pool. Think for example a car assembly process. It requires a number of wheels, a number of doors and a chassis. Each of these elements are produced independently according to their process model definitions. Thus, there can exist three pools, one with "CarWheel" instance type, one with "CarDoor" type and one with "CarChassis" type. The selector attribute is used for discriminating instances that are of the same type, yet of a different variant. For example, the "CarDoor" instance type can have the color (e.g., blue/red) as attribute. A pool is a virtual "container" to keep homogenous process instances;

homogeneous from an instance type perspective, as these can have different attributes. All these concepts are illustrated in Figure 66. Process instances are denoted as shape figures.

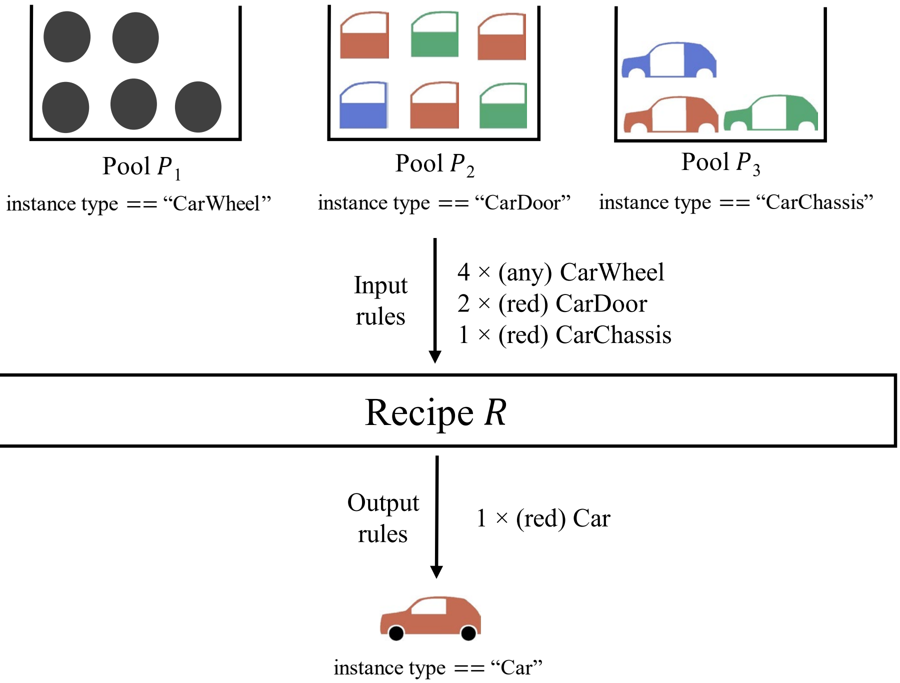


Figure 66: Illustration of the recipe concepts trough an example.

The configuration of each pool plays a crucial role for the fulfilment of a recipe. The following options are considered:

- **Genericity**. A pool can be either *generic* or *specific*. In the first case, the pool does not consider the selector attribute of the buffered process instances (e.g., in Pool $\mathcal{P}_1$ of Figure 66). In the latter case, recipe fulfillment candidates are nominated based on the selector attribute (e.g., on the color attribute in pools $\mathcal{P}_2$ and $\mathcal{P}_3$ of Figure 66).
- **Availability mask**. Pools can represent physical buffers but as such should account for physical availability, i.e., how instances/objects are accessed. This research considers three availability masks:
  - ALL: All instances are available (e.g., in a virtual or physical pool that physical layout is not relevant).
  - FIRST: The instance that was first placed in the pool is considered as available. Subsequent instances are marked as available if and only if they share the selector attribute value of the first instance, in one sustained sequence.
  - LAST: The instance that was placed last in the pool is considered as available. Subsequent instances are marked as available if and only if they share the selector attribute value of the last instance, in one sustained sequence.

- **Release policy**. The release policy ranks instances for recipe fulfillment (and thus "release" from the pool). This research considers three policies:
    - FIFO: instances that have been in the pool the longest are released first.
    - LIFO: instances that have been in the pool the shortest are released first.
    - ATTR: instances are released based on a selector attribute value.
- **Fulfillment cardinality**. The fulfillment cardinality determines how many instances of a pool are needed to lead to recipe fulfillment. It can be a single value, i.e., all instances are nominated for fulfillment, or it can take a minimum (n) and a maximum (m) value, i.e., the pool needs at least n and less than m.

With the configuration options described above, a recipe can be specified with the following notation, shown in Table 6 (based on the example of Figure 66). Upon a recipe fulfilment, a process may continue its flow after the respective synchronization points or a new process instance (mainly from a different process definition) can start.

Table 6: Specification of a recipe through an example.

Recipe name:	Final car assembly					
Selector attribute:	ordernumber					
Input instance type	min	max	gen	relpol	mask	rel
CarWheel	4	4	●	FIFO	LAST	○
CarDoor	4	4	○	LIFO	ALL	○
CarChassis	1	1	○	LIFO	ALL	●
num	Start process definition key (output)					
1	Final_Car_Assembly_Process					

Based on the proposed notation, recipes are constructed for the manufacturing constructs under concern. Table 7 shows an illustrative example of a recipe representing the buffering construct. The recipe represents a single buffer (physical or virtual) that keeps exactly 10 items (physical items or virtual entities).

Table 7: Example of a recipe representing buffering.

Recipe name:	Buffer 10 items					
Selector attribute:	None					
Input instance type	min	max	gen	relpol	mask	rel
Item	10	10	●	FIFO	FIRST	●

The example shown in Table 6 represents a recipe for the bundling construct, i.e., various parts are merged together. A more dynamic example of a bundling recipe is shown in Table 8. A worker stacks items in a cart for transportation in a storage zone. Once ten items are placed in the cart, he can then transport them.

Table 8: Example of a recipe representing bundling.

Recipe name:	Transporting up to 10 items on a cart					
Selector attribute:	None					
Input instance type	min	max	gen	relpol	mask	rel
Item	1	10	●	LIFO	FIRST	●
Cart	1	1	●	FIFO	FIRST	●

Table 9 shows the recipe for the counteractivities of the recipe of Table 8. Once the cart with the items has been transported at the storage zone, item have been unloaded. To ensure that the items from various carts are not mixed, a specific selector attribute (from the recipe of Table 8) is defined.

Table 9: Example of a recipe representing unbundling.

Recipe name:	Final car assembly					
Selector attribute:	Fulfillment identifier from recipe "Transporting up to 10 items on a cart"					
Input instance type	min	max	gen	relpol	mask	rel
Item	1	10	○	FIFO	ALL	●
Cart	1	1	○	FIFO	ALL	●

Regarding the recipe process notation, Figure 67 shows an example of how recipes can be defined onto process models.

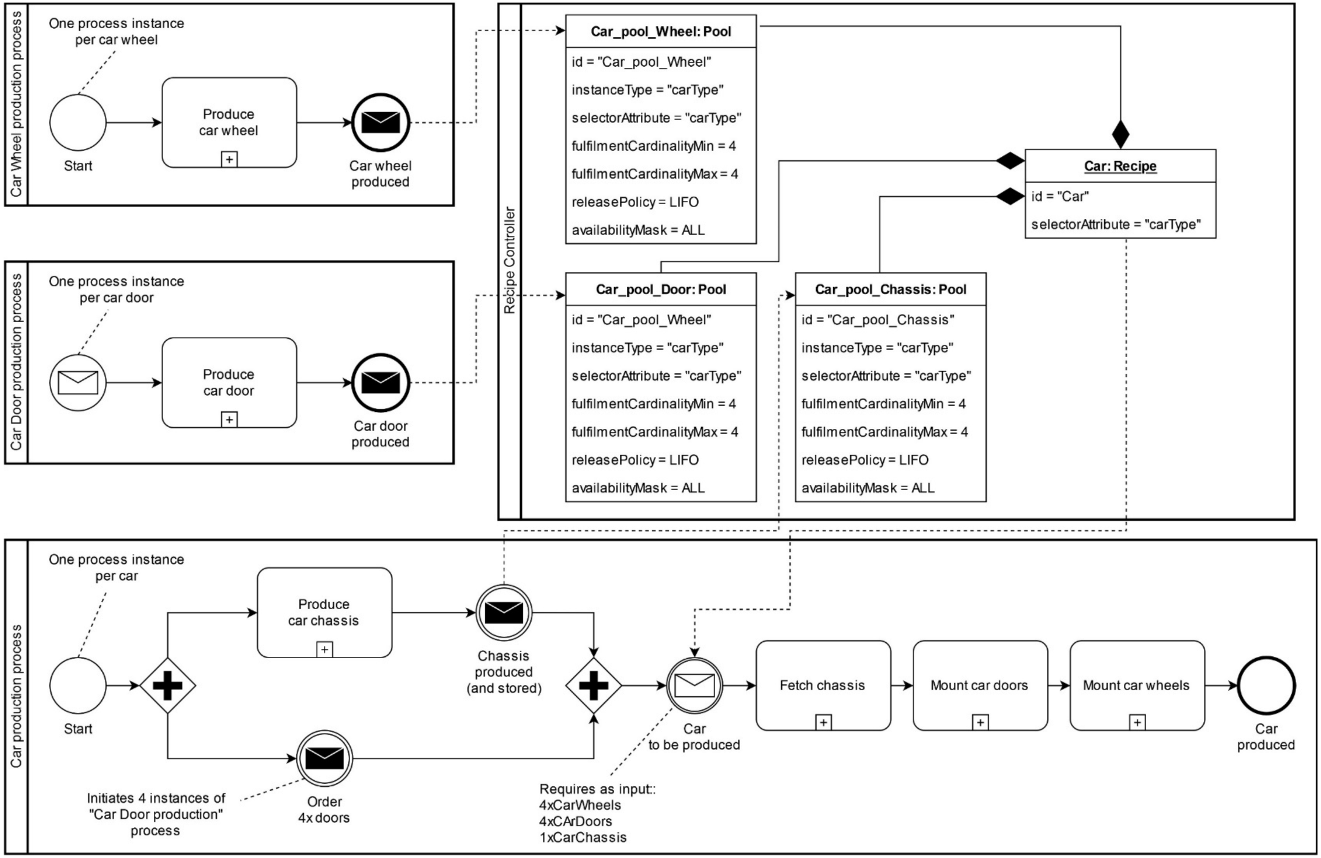


Figure 67: Illustrative example for the proposed recipe process notation.

The notation is a combination of BPMN elements and UML class diagrams. As the main goal is to provide a way to depict recipes onto process models, i.e., the intended functionality, the aesthetics are of less importance. While such notation can serve communication and collaboration purposes, it should be also noted that the are no execution semantics. At the current state, transformations from the models to digital entities are missing and recipes have to be inserted as digital entities in a manual way through its technical operationalization (discussed in Section 6.2.1).

3.5.3.2 Formalization of recipe concepts

Recipes ($\mathcal{R}$) are treated as sequences that contain Pools ($\mathcal{P}$) that are treated as sequences that contain instances. The notation $|\mathcal{P}|$ is used to denote the number of instances currently in pool $\mathcal{P}$. The notation $\mathcal{P}(i)$, with $i \in \{1, \dots, |\mathcal{P}|\}$, refers to the i-th instance in the pool. Not to be confused with the powerset notation $\mathcal{P}(A)$, referring to the powerset of set A. Note that this instance indexing is based on the time at which an instance was added to the pool. In other words, from a mathematical perspective, a pool is an array of instances that is sorted on arrival timestamp. In general, the symbol i is used to either denote an array index (like in the $\mathcal{P}(i)$ notation) or a process instance, like $i \in \mathcal{P}$. The latter should be read as *instance i in pool $\mathcal{P}$*. The mathematical model, which extends the content presented in the previous section, uses the following symbols:

$\mathcal{R}$ a recipe.
$\mathcal{P}$ a pool. Is a member of a recipe, i.e., $\mathcal{P} \in \mathcal{R}$.
$\mathcal{S}$ the (abstract) set of possible selector attributes.
$s_{\mathcal{P}}$ the selector attribute for pool $\mathcal{P}$.
$\mathcal{V}_s$ the (abstract) set of possible selector attribute values for selector attribute $s \in \mathcal{S}$.
v_i the selector attribute value for instance $i \in \mathcal{P}$.
$c_{\mathcal{P}}^-$ the minimum fulfilment cardinality for pool $\mathcal{P}$.
$c_{\mathcal{P}}^+$ the maximum fulfilment cardinality for pool $\mathcal{P}$.
$\alpha_{\mathcal{P}}(i)$ availability mask function for pool $\mathcal{P}$. $\alpha_{\mathcal{P}}(i) \in \{0,1\}\ \forall i \in \mathcal{P}$.
$\rho_{\mathcal{P}}(i)$ release policy ranking function for pool $\mathcal{P}$. $\rho_{\mathcal{P}}(i) \in \{1, \dots, |\mathcal{P}|\}\ \forall i \in \mathcal{P}$.
$g_{\mathcal{P}}$ boolean whether pool $\mathcal{P}$ is generic (1) or specific (0). $g_{\mathcal{P}} \in \{0,1\}$.
$\mathcal{S}(\mathcal{P})$ the set of selector attribute values for which at least $c_{\mathcal{P}}^-$ instances exist in pool $\mathcal{P}$. Formally defined as

$$\mathcal{S}(\mathcal{P}) \quad \equiv \{v \in \{v_p : p \in \mathcal{P}\} : |\{v_p : p \in \mathcal{P} \wedge v_p = v\}| \geq c_{\mathcal{P}}^-\} \quad (1)$$

Note that, by definition, $\mathcal{S}(\mathcal{P}) \subseteq \mathcal{V}_{s_{\mathcal{P}}}$ holds.

$m(\mathcal{P})$ a map that maps an attribute value to a sequence of fulfilment candidate instances (of the same attribute value) in pool $\mathcal{P}$.

$$m(\mathcal{P}) : v \rightarrow I \quad (2)$$

with $v \in \mathcal{S}(\mathcal{P})$ and set of instances $I \subseteq \mathcal{P}$.

Later in the discussion, Figure 68 introduces an example of such a mapping.

3.5.3.2.1 Availability mask functions

Availability masking uses a boolean mask to indicate whether an instance is available for recipe fulfilment. The mask $\alpha_{\mathcal{P}}(i)$ equals to 1 if and only if the instance argument i is available for recipe fulfilment (otherwise 0). Consequently, an instance may only be nominated for a fulfilment if $\alpha_{\mathcal{P}}(i) = 1$ holds for instance $i \in \mathcal{P}$. There are three flavors of availability masks. First, there is the `ALL` mask, which means that all instances are available.

Alternatively, there is the `FIRST` mask, which marks the first element as available. Subsequent instances are available if and only if they share the selector attribute value of the *first* instance, in one sustained sequence (as is often the case in physical stacks only accessible from the stacking direction). Somewhat inversely, there is the `LAST` mask. As the name suggests, this mask marks the last element as available. Preceding instances are available if and only if they share the selector attribute of the *last* instance, in one sustained sequence. All three masks are defined with the following equations:

$$\alpha_{\mathcal{P}}^{\text{ALL}}(\mathcal{P}(i)) \equiv 1, \qquad \forall i \in \{1, \dots, |\mathcal{P}|\} \quad (3)$$

$$\alpha_{\mathcal{P}}^{\text{FIRST}}(\mathcal{P}(i)) \equiv \begin{cases} 1 & \text{if } i = 1 \vee (v_{\mathcal{P}(i)} = v_{\mathcal{P}(i-1)} = \dots = v_{\mathcal{P}(1)}) \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in \{1, \dots, |\mathcal{P}|\} \quad (4)$$

$$\alpha_{\mathcal{P}}^{\text{LAST}}(\mathcal{P}(i)) \equiv \begin{cases} 1 & \text{if } i = |\mathcal{P}| \vee (v_{\mathcal{P}(i)} = v_{\mathcal{P}(i+1)} = \dots = v_{\mathcal{P}(|\mathcal{P}|)}) \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in \{1, \dots, |\mathcal{P}|\} \quad (5)$$

3.5.3.2.2 Release policy functions

Release policies use a ranking function to prioritize instances for fulfilment. A lower rank means the instance is preferred. First off, there is the First-In-First-Out (`FIFO`) release policy, which orders instances based on the timestamp t at which they were added to the recipe pool.

$$i_1 \prec i_2 \quad \Leftrightarrow t_{i_1} \le t_{i_2}, \qquad \forall (i_1, i_2) \in \mathcal{P} \times \mathcal{P} \quad (6)$$

Instance i_1 is preferred over i_2 for release, if and only if the time added to the pool of i_1, t_{i_1} is smaller than or equal to that of i_2, t_{i_2}. In other words: the instances are ranked such that their timestamps are non-decreasing. The ranking function, $\rho_{\mathcal{P}}^{\text{FIFO}}$ is therefore defined simply as the instance index of the time-sorted sequence of instances in a pool:

$$\rho_{\mathcal{P}}^{\text{FIFO}}(\mathcal{P}(i)) \quad \equiv i, \qquad \forall i \in \{1, \dots, |\mathcal{P}|\} \qquad (7)$$

Secondly, there is the inverse of `FIFO`, Last-In-First-Out (`LIFO`), again based on timestamp t.

$$i_1 \prec i_2 \quad \Leftrightarrow t_{i_1} \ge t_{i_2}, \qquad \forall (i_1, i_2) \in \mathcal{P} \times \mathcal{P} \quad (8)$$

Notice that Eq. (8) results in the reverse ranking of Eq. (6). The resulting ranking function, $\rho_{\mathcal{P}}^{\text{LIFO}}$ is therefore the inverse ranking of Eq. (7):

$$\rho_{\mathcal{P}}^{\text{LIFO}}(\mathcal{P}(i)) \quad \equiv 1 + |\mathcal{P}| - i, \qquad \forall i \in \{1, \dots, |\mathcal{P}|\} \quad (9)$$

Lastly, there is the attribute based policy (`ATTR`), which sorts instances based on some attribute, denoted by #. As an instantiation example of this policy, one could think of a priority based policy.

$$i_1 \prec i_2 \quad \Leftrightarrow \#_{i_1} \ge \#_{i_2}, \qquad \forall (i_1, i_2) \in \mathcal{P} \times \mathcal{P} \quad (10)$$

To define the ATTR release policy ranking function, we first define the sequence $\text{sort}^{\downarrow}(A, \#) \subseteq A$ to be the result of sorting sequence A on some attribute # in descending order (i.e. the result is nonincreasing). Furthermore, we define $\text{index}(i, A) \in \{1, \dots, |A|\}$ to return the index at which element i occurs in sequence A. Using these intermediate definitions, we can arrive at the final definition:

$$\rho_{\mathcal{P}}^{\text{ATTR}}(\mathcal{P}(i)) \quad \equiv \text{index}\big(\mathcal{P}(i), \text{sort}^{\downarrow}(\{\mathcal{P}, \#\})\big), \qquad \forall i \in \{1, \dots, |\mathcal{P}|\} \quad (11)$$

where # refers to the priority attribute to be sorted.
Given the properties of these functions, the discussion above can be generalized to

$$i_1 \prec i_2 \quad \Leftrightarrow \rho_{\mathcal{P}}(i_1) \le \rho_{\mathcal{P}}(i_2) \qquad \forall (i_1, i_2) \in \mathcal{P} \times \mathcal{P} \quad (12)$$

This generalized form is used in the subsequent implementation. The function definition denoted by $\rho_{\mathcal{P}}$ is to be replaced with an appropriate release policy function variant.
Note that, since output rules are released instantaneously once a recipe is fulfilled, the effect of these release policies is only observable if there is a choice which instances should remain in the pool. This choice is only there if there are more instances in the pool than the maximum fulfilment cardinality, i.e., $|m(v \in \mathcal{S}(\mathcal{P}))| > c_{\mathcal{P}}^{+}$. Otherwise, exactly $\min(c_{\mathcal{P}}^{+}, |\mathcal{P}|)$ instances are selected in the fulfilment and the ordering is irrelevant, as becomes apparent in the following algorithmic discussion.

3.5.3.2.3 The Pool algorithm

As mentioned before, a pool can produce a mapping $m: v \in \mathcal{S}(\mathcal{P}) \to I \subseteq \mathcal{P}$ upon request. This mapping maps an attribute value v to a sequence of fulfilment candidate instances I. A visual example that explains how that mapping works, can be found in Figure 68. In this figure, the "CarDoor" pool from Figure 66 is used as an example.

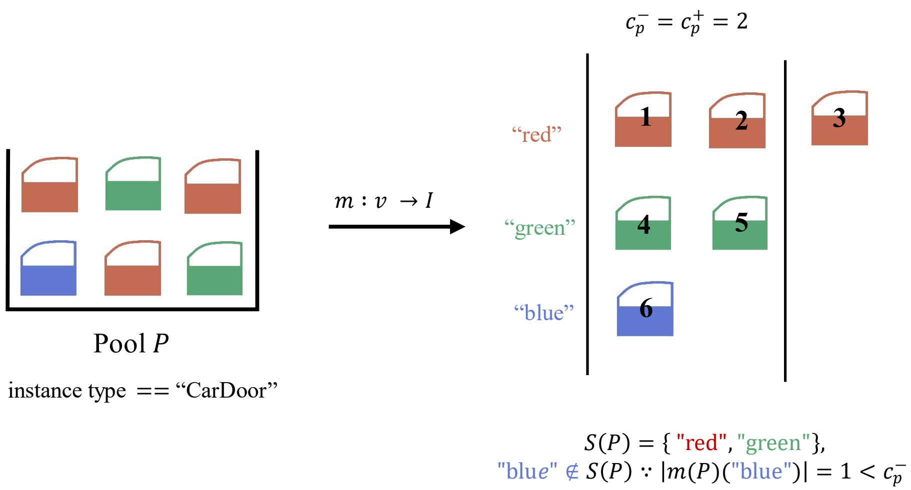


Figure 68: Map generation $m(\mathcal{P})$ example.

The pool's mapping algorithm is listed in Figure 69.

Algorithm 1: Pool's mapping algorithm, i.e. $m(\mathcal{P})$.

Input: Pool $\mathcal{P}$.

Output: Mapping of attribute values to sequence of fulfillment candidate instances, $m : v \in \mathcal{S}(\mathcal{P}) \rightarrow I \subseteq \mathcal{P}$.

/* Map generation phase. */

$m_1 \leftarrow (\{\} \rightarrow \{\})$; /* Initialize empty map m_1. */

foreach $i \in \{i \in \mathcal{P} : \alpha_{\mathcal{P}}(i) = 1\}$ **do** /* For every available instance i in the pool. */

  if $m_1(v_i) = \varnothing$ **then** /* If value v_i not in map m_1 yet. */

    $m_1(v_i) \leftarrow \{\}$; /* Add new value v_i to map m_1. */

  end

  $m_1(v_i) \leftarrow m_1(v_i) \cup \{i\}$; /* Add instance i to map m_1. */

end

/* Map pruning phase. */

$m_2 \leftarrow (\{\} \rightarrow \{\})$; /* Initialize empty map m_2. */

foreach $v \in m_1$ **do** /* For every key value in map m_1. */

  if $|m(v)| \geq c_{\mathcal{P}}^{-}$ **then** /* At least $c_{\mathcal{P}}^{-}$ instances exist for value v. */

    $m_1(v) \leftarrow \text{sort}^{\uparrow}(m(v),\ \rho_{\mathcal{P}})$; /* Rank instances based on release policy. */

    $l \leftarrow \{\}$; /* Initialize empty candidate list. */

    $x \leftarrow \min\left(|\mathcal{P}|,\ c_{\mathcal{P}}^{+}\right)$; /* Determine how many instances to nominate. */

    for $(i \leftarrow 1; i \leq x; i \leftarrow i + 1)$ **do** /* For every nominated instance. */

      $l \leftarrow l \cup \{m_1(v)(i)\}$; /* Add instance $m_1(v)(i)$ to list of candidates. */

    end

    $m_2(v) \leftarrow l$; /* Place list of candidates in pruned map m_2. */

  end

end

return m_2; /* Return the pruned map m_2. */

Figure 69: Pool's mapping algorithm.

3.5.3.2.4 The Recipe algorithm

The recipe algorithm collects and analyses pool maps to determine fulfilment feasibility. If a fulfilment can be achieved for a particular selector attribute value, the algorithm releases the appropriate instances from the pools and returns them in a list. The algorithm is listed in Figure 70.

Algorithm 2: Recipe's fulfillment algorithm.

```
Output: Sequence of buffered instances that are part of the fulfillment. Empty
        sequence if recipe cannot be fulfilled.
/* Map analysis phase.                                                        */
v ← ∅;                              /* Initialize sequence of potential fulfillment values. */
foreach p ∈ R do                                          /* For each pool in recipe. */
    m_p ← m(p);                                  /* Query and store the pool's map. */
    if |keys(m_p)| = 0 ∧ c_p^- ≠ 0 then           /* If this pool cannot be fulfilled. */
        v ← ∅;                                  /* A global fulfillment is infeasible. */
        break;
    end
    if v = ∅ then                              /* If this is the first pool to analyze. */
        /* Take the first pool's potential fulfillment values as starting point.   */
        if c_p^- = 0 then
            v ← {∅};             /* Add generic null value as potential fulfillment value. */
        else
            v ← keys(m_p);            /* Add potential fulfillment values to sequence. */
        end
    end
    if g_p = 0 ∧ c_p^- ≠ 0 ∧ |p| ≠ 0 then        /* If this pool should be accounted for in
     fulfillment feasibility. */
        if v = {∅} then     /* If the previous pool was a generic pool (or was a satisfied
         pool with 0 candidates), but this pool is not. */
            v ← keys(m_p);                     /* Overwrite potential fulfillment values. */
        else
            v ← v ∩ keys(m_p);                     /* Prune potential fulfillment values. */
        end
    end
end
/* Fulfillment feasibility analysis phase.                                    */
if v = ∅ ∨ |v| = 0 then                             /* If no fulfillment is feasible. */
    return {};                                          /* Return empty sequence. */
end
f ← v(1);                     /* Pick the or a fulfillment value and store it in f. */
r ← {};                                /* Initialize sequence of released instances. */
/* Note: f = ∅ can hold true by design, in case of a generic fulfillment.     */
/* Data restructure phase.                                                    */
foreach p ∈ R do                                                 /* For each pool. */
    if ¬(c_p^- = 0 ∧ |m(p)| = 0) then                      /* Skip empty optional pools. */
        foreach i ∈ m(p)(f) do                  /* For every to-be-released instance. */
            release(p, i);                              /* Release instance from pool. */
            r ← r ∪ {i};                    /* Add instance i to sequence of released. */
        end
    end
end
return r;                                 /* Return sequence of released instances. */
```

Figure 70: Recipe's fulfilment algorithm.

3.6 Chapter conclusion

High mix-low volume production environments and introduction of various heterogeneous technologies to perform production operations result in a high degree of complexity in production processes. Modeling support is necessary to both depict correctly the desired operations but also to ease their enactment. This chapter provides such support by designing modeling constructs and mechanisms that can be used to model manufacturing processes in BPMN.

After studying literature for the use and suitability of the notation in (smart) manufacturing, and its limitations, three artefacts are designed to support modelers: 1) constructs for task assignment to heterogeneous agents, 2) constructs for modeling human-robot collaborative processes, and 3) an activities synchronization mechanism to support manufacturing constructs such as buffering and bundling/unbundling. The design of the artefacts considers the need that the constructs should support the execution of the modeled processes. Their operationalization is discussed in Chapter 6.

The most complex artefact, the synchronization mechanism, has been evaluated on its utility, in the frames of a master thesis project (Spijkers, 2019), which the author of the current thesis was guiding and supervising. Through a workshop in which the mechanism was explained, eight practitioners were asked to give their opinion on perceived ease of use (PEoU) and perceived usefulness (PU) (Davis, 1989). The evaluation panel perceived the method as useful in modeling situations that require synchronization, which they find an interesting and relevant topic. Moreover, insights were gained on further research and extension points (discussed in Section 8.4). However, conclusions should be treated with caution as the practitioners had rather limited experience with BPMN.

CHAPTER 4

Exception handling

The dynamicity of (market, business, and manufacturing) environments and the plethora of technologies encountered in smart factories have given rise to exceptional situations, as thoroughly discussed in Section 2.1.6.2. Such situations can be an order cancellation, a machine breakdown or a system failure, which regardless if these can be expected or not, they have negative impact on an organization's performance and costs (Bruccoleri et al., 2007).

The term exception has seen many definitions. Luo et al. (2000) view exceptions as *facts or situations that are not modeled by the information systems or deviations between what we plan and what actually happens*. *The differences between the actual and the expected state of a production system* are considered as *exceptions* by Bruccoleri et al. (2003), as well. Russell et al. (2006b) term the *deviations from normal execution arising during a business process* as exceptions, while similarly, Andree et al. (2020) describe *a discrepancy of a business process between the planned flow and the reality* as an *exception*. A bit differently, Lohmeyer (2013) considers the success or the failure of the goal of a business process as the differentiating point to consider a deviation as an exception or not. If a goal is ultimately achieved, then any deviations from a "normal" sequence of steps is considered as an *alternate flow* and not as exceptions. While the above definitions are heavily process/workflow-oriented, exceptions can happen on a system or a single unit level, like for example an alert that a temperature sensor raises. For the interest of this research, exceptions from individual units are studied with respect to their underlying process(es). In other words, exceptions are not studied as individual entities/objects, but as part of processes in concern.

Exceptions can be characterized along different dimensions. Luo et al. (2000) use an orthogonal, three-dimensional space to analyze exceptions:

- *Known* dimension, that distinguishes between known and unknown, i.e., those that the system has met before or not. A *learning* process can make the unknown exceptions known. This dimension is also characterized in literature as the distinction between expected (or anticipated) and unexpected (unanticipated) exceptions (Reichert & Weber, 2012a).

- *Detectable* dimension, that distinguishes between detectable and undetectable exceptions, depending on the capability of a system to notice the occurrence of an exception or not. Detection can be achieved through supervision of the workflow system's external environment and comparison with its specified behavior.

- *Resolvable* dimension, that distinguishes between resolvable and irresolvable exceptions, depending on whether the system can derive or not a solution with exception handling mechanisms.

The current research aims to support the handling of exceptions, i.e., to make them resolvable, either known or unknown ones. The detection dimension is not under concern and is rather taken as assumption that there are systems to detect any deviating behavior (e.g., a situation awareness system). Of course, unknown exceptions are usually undetectable, but the system should cater for resolving the issues to avoid downtimes, extra costs, and/or performance deterioration. The characteristics of the exceptions under concern are illustrated in Figure 71, within the three-dimensional exception knowledge space of Luo et al. (2000). Moreover, as the current research considers a process-oriented view on tackling complexity on manufacturing operations, an *exception* is defined as "*any event that disrupts the normal behavior of the designed manufacturing operations*".

"Exception" definition

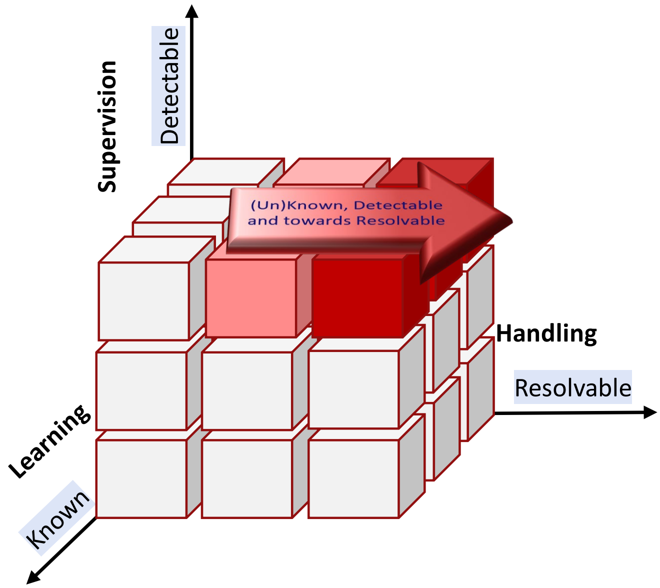


Figure 71: Characteristics of exceptions that current research treats, in the three-dimensional space of Luo et al. (2000).

Apart from the three dimensions that discussed above, other characteristics of exceptions are useful to consider, especially when these should be treated by a BPM system (Kurz et al., 2013): arbitrary time of occurrence (if any), response actions depend on the state of other processes, and necessity for IT support. Especially for the third one, while automated way of handling exceptions is usually preferred, the human involvement cannot be avoided and often can be catalytic.

Regardless of where the exceptions originate from or what type they are, exception handling, i.e., the process of reacting and taking corrective actions upon occurrence (Bruccoleri et al., 2003), is essential for an organization to eliminate or reduce the negative impact on the business (Milliken, 2011) and eventually remain competitive (Grauer et al., 2010). However, the diversity of exceptions in dynamic environments is a challenge, especially on operational level, where complexity of processes and strict time constraints demand for substantive and fast reaction (Wang et al., 2020). Structured exception handing guidelines are necessary to describe and connect the process from detection to correct resolution.

Thus, the purpose of this chapter is to provide an answer on how occurred exceptions can be handled in dynamic manufacturing environments (RQ3) by designing exception handling guidelines for identified types of exceptions. As the focus of this research is on MOM level (Figure 7), the guidelines are scoped onto the operational level of decision, in the short-term timeframes, leaving out of scope decisions on tactical or strategic level that mostly apply in mid/long-term time frames.

4.1 Chapter outline

The answer to RQ2 is provided through design science research (similarly as for RQ1 in Chapter 3), as illustrated in Figure 72 (which zooms-in the aspect under concern of the three-aspects design and development of Figure 13). The academic rigor is safeguarded through a systematic literature review (SLR) on categorizing the types of exceptions that (can) occur in manufacturing environments. Scientific literature is also consulted for identifying the strategy values and the KPIs that affect the exception handling guideline. The practical relevance is secured by analyzing exceptions that occur in real-world operations environments. More specifically, a data source with information on exceptions was provided by the TRI pilot case of HORSE project and qualitative interviews were conducted with three pilot cases of SHOP4CF project. The results of the analysis from the "environment" are consolidated with the results from the SLR to generate (first design activity) a categorization of exception types. The categorization is used as input for the second design activity, which yields, as an artefact, a set of guidelines for operational exception handling.

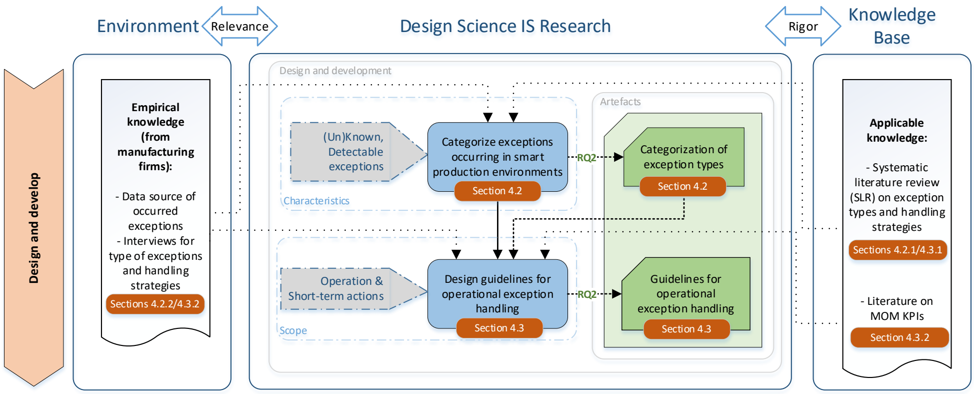


Figure 72: DSR approach for RQ2 – Exception handling in dynamic manufacturing environments.

This chapter is organized according to the DSR approach shown in Figure 72. Section 4.2 presents the design of the categorization of exception types, through the SLR and the analysis of inputs from practice. Section 4.3 presents the design of the exception handling guideline. The chapter is concluded in Section 4.4.

4.2 Categorization of exception types

The way to interact on exceptions depends on the type of exception. Thus, it is important for an organization to have a clear picture of the types of exceptions that might appear in their environments. Accordingly, the strategies and methods to handle exceptions should be clear as well. As this research aims to provide guidance on handling operational exceptions, it is therefore essential to first categorize the exceptions. In this section, the first of the two research activities of the DSR approach of Figure 72 (top blue box) is discussed. The methodology that is followed to lead in a categorization of exceptions is illustrated in Figure 73 and summarized below.

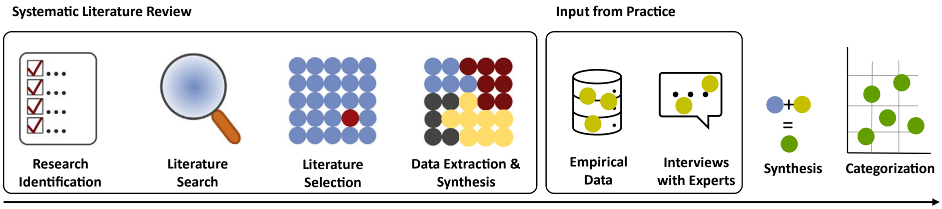


Figure 73: Methodology for categorization of exception types (adapted from (Leitner & Rinderle-Ma, 2014)).

- A systematic literature review is performed to identify types of exceptions appearing in the scientific knowledge base. A thorough, sophisticated literature review is the foundation and inspiration for substantial and useful research[48] (Boote & Beile, 2005). The SLR is discussed in Section 4.2.1.
- Moreover, input from practice is gathered to identify any exceptions that do not appear in literature but are encountered in real-world scenarios. Two kinds of practical inputs were gathered: i) empirical data that represent occurred exceptions at one of the practical pilot cases (more specifically the TRI pilot), ii) discussion through semi-structured interviews with practitioners from the pilots. The practical inputs are discussed in Section 4.2.2.
- Both scientific and practical inputs are consolidated to lead into the categorization of exceptions. The synthesis and design are elaborated in Section 4.2.3.

4.2.1 Systematic literature review (SLR) on "Exceptions"

The SRL is conducted through four main steps (as shown in Figure 73):

i. Research Identification
ii. Literature Search
iii. Literature Selection
iv. Data Extraction and Synthesis

i) <u>Research identification</u>

The objective is to identify types of exceptions that occur in manufacturing environments and how they are categorized, so as the handling of those is more structured. Thus, the main research question to be answered is the following:

How are exceptions categorized in manufacturing domain and what are the typical handling methods?

---

48 "*A researcher cannot perform significant research without first understanding the literature in the field*" (Boote & Beile, 2005, p.3).

ii), iii) Literature Search & Selection

Looking for existing work that have dealt with the above research question is performed on databases/search engines that include a broad base of scientific studies. ScienceDirect[49] and Scopus[50] are considered adequate sources for the field under consideration. ScienceDirect exclusively covers journal articles, while Scopus also covers conference proceedings.

Regarding the search term(s) that shall bring potentially relevant and interesting results, synonyms and similar terms to the term "exception" should be considered. For instance, the terms "deviation" or "error" are often encountered to describe disturbances in processes. Similarly, as the exceptions (types) can be "classified" or "taxonomized", extra terms have been considered apart from the "categorization" term. Furthermore, an incremental refinement is necessary to find a reasonable number of results. The iterative process of search terms is listed on Table 10. Of course, as each search engine has its own search capabilities and syntactical requirements, the search terms are adapted accordingly. The detailed search terms with individual results are presented on Appendix D.1.

Table 10: Incremental refinement steps of the SLR search term(s).

Search term	Reasoning
Exception AND (type OR categor* OR classification OR taxonomy OR pattern OR handling)	Exploratory search term including only the term "exception" (no synonyms or relevant alternatives).
(exception* OR failure OR error OR deviat* OR defect) AND (type OR categor* OR classification OR taxonomy OR pattern OR handling)	Including other terms (i.e., failure, error, deviation, defect) that often appear in literature and practice to represent "disturbances" from on objective.
((exception* OR failure OR error OR deviat* OR defect) AND (type OR categor* OR classification OR taxonomy OR pattern OR handling)) AND NOT (medic* OR *coding OR code OR programming OR neural OR optical OR training)	Excluding specific terms that further limit down irrelevant studies. For instance, excluding studies on medicine/healthcare domain or studies referring to neural networks.
((exception* OR failure OR error OR deviat* OR defect) AND (type OR categor* OR classification OR taxonomy OR pattern OR handling)) AND (process OR *flow) AND NOT (medic* OR *coding OR code OR programming OR neural OR optical OR training)	Scoping results towards "process" or "workflow" domain.
((exception* OR failure OR error OR deviat* OR defect) AND (type OR categor* OR classification OR taxonomy OR pattern OR handling)) AND (manufacturing OR process OR *flow) AND NOT (medic* OR *coding OR code OR programming OR neural OR optical OR training)	Adding also "manufacturing" domain for more specific scoping.
((exception* OR failure OR error OR deviat*) AND (type OR categor* OR classification OR taxonomy OR pattern OR handling)) AND	From a quick scanning, the term "defect" resulted in studies that have a narrow scope (e.g., on product quality) than the current research. So,

[49] https://www.sciencedirect.com/

[50] https://www.scopus.com

(process OR *flow) AND NOT (medic* OR *coding OR code OR programming OR neural OR optical OR training)	the term "defect*" was removed. Given that the rest terms provide a lot of coverage, this exclusion should not have great impact on rigor.
((exception* OR failure OR error OR deviat*) AND (type OR categor* OR classification OR taxonomy OR pattern OR handling)) AND (process OR *flow) AND NOT (medic* OR *coding OR code OR programming OR neural OR optical OR training OR simulat* OR {type I error} OR {type II error})	Excluding studies focusing on simulation or mathematical problems.

Inclusion/exclusion criteria are specified to further limit the search results:

- Fetch studies that are published only the last 15 years before the SLR is conducted (i.e., excluding studies from 2005 and before). 15 years is considered a representative period for state-of-the-art work on the topic. Of course, with the "snowballing method (Wohlin, 2014), relevant work published before 2005 can be obtained.
- Studies should be published in English language.
- Studies for which the full-text is accessible are considered.

The search and selection procedures are illustrated in Figure 74. The results from both Scopus and ScienceDirect are first checked for duplicates. A first filtering is performed based on the relevance of the title, leading in a "long list" of results. At the next step, the remaining results are filtered on relevance based on both title and abstract, leading to a "short list". With "snowballing", extra studies are retrieved through Google Scholar[51] that are missing from the results. The final short list of selected studies to be analyzed is available on Appendix D.2.

[51] https://scholar.google.com/

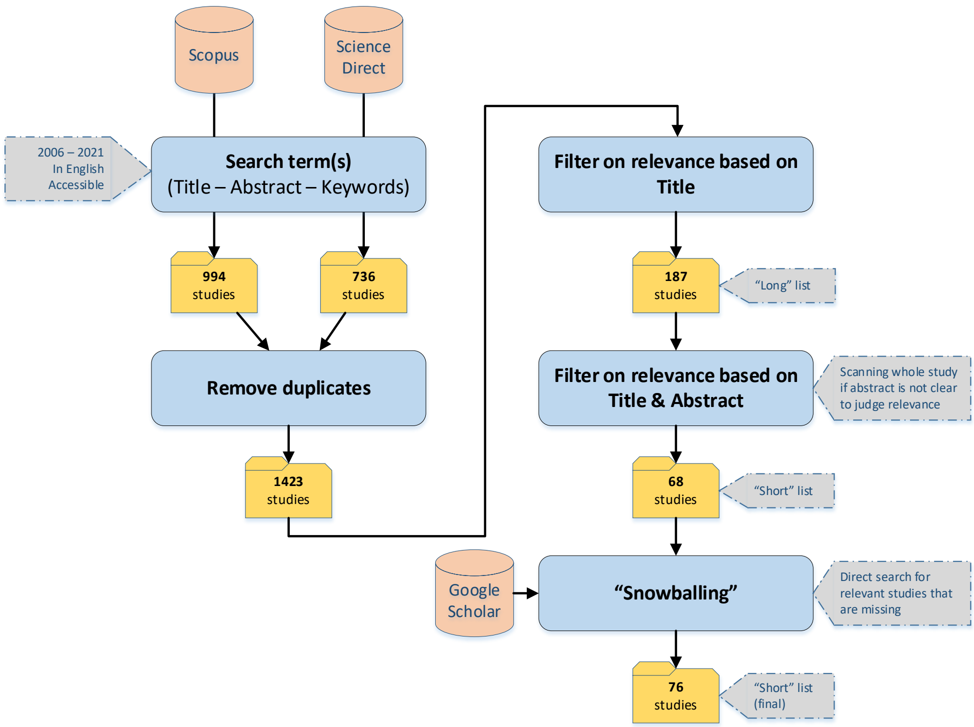


Figure 74: SLR search and selection procedure overview.

iv) Data Extraction and Synthesis

From the final short list of selected studies, 19 studies yielded types of exceptions. All those types have been categorized under common groups (by logical reasoning) to obtain a reasonable number of categories. Table 11 presents the results.

The categories are rather self-explanatory, so no more analysis is required. It is noteworthy to mention though, that some studies study exceptions from a very general or high-level perspective, e.g., machine-related or process-related exceptions by Keddis et al. (2016), while others present more detailed categories, e.g., wear on tools by Qian et al. (2020). Moreover, the study from Eder & Liebhart (1995) simply discusses the fact whether an exception is expected or not. This "type" has been placed in the "Various" categories at this step. In the synthesis phase (discussed in Section 4.2.3), it is revised whether it is included in the designed categorization or not.

Table 11: Categorization of exception types through analysis of SLR.

Exception types categories			Studies																		
			Dahl et al. (2021)	Chavez et al. (2020)	Qian et al. (2020)	Farooqui et al. (2016)	Keddis et al. (2016)	Sahno et al. (2015)	Ritter & Sosulski (2014)	Reichert & Weber (2012)	Antunes (2011)	Misic et al. (2010)	de Snoo et al. (2010)	Adams et al. (2007)	Bruccoleri et al. (2007)	Russell et al. (2006a)	Mourão & Antunes (2005)	Luo et al. (2000)	Casati et al. (1999)	Eder & Liebhart (1995)	Strong & Miller (1995)
Resource -related		General										X									
	Machine-related	General					X	X													
		Equipment malfunction	X		X	X															
		Machine/equipment breakdown		X	X	X									X						
		Limited operations/Deterioration			X										X						
		Task failure (expected/unexpected)	X																		
		Specification errors	X																		
		Unavailable								X				X			X				
	Tool-related	Wear			X										X						
		Disqualified			X																
		Occupied			X																
	Material/ Product-related	Unavailable		X	X					X				X			X				
		Delayed delivery			X							X									
		Missing parts			X	X															
		Processability issues			X																
		Quality issues		X	X																
	Personnel-related	Operator absenteeism/Unavailability		X	X			X		X				X			X				
		Disqualified			X																
		Improper operation			X																
		Delayed operations			X							X									
		Training deficiency						X													
		Change in orders (requirements)		X	X								X								

Order-related		Cancellation			X								X								
		Priority change/Rush orders			X								X		X						
Process-related		General					X	X										X	X		
		Activity failure							X	X											
		Message-flow failure							X												
		Operation errors										X									
		Work item abort/failure							X	X				X		X					
	Time-related	General												X		X					
		Timeouts/Deadline expiration																			X
Event-related		General									X										
		Temporal events															X		X		
		Workflow events															X		X		
		External events							X	X				X		X	X		X		
		Unspecified events	X																		
Data-related		General										X							X		
		Application failures (unexpected data)									X							X		X	
Various		Software bugs				X															
		Constraint rules												X							
		Constraint violations							X	X				X		X					
		Communication failure	X																		
		Basic failures									X									X	
		Infrastructure exception																X			
		Power outage				X															
		Management problem						X													
		Supplier/subcontractor problem						X													
		Design problem/error						X													X
		Disqualified working conditions			X																
		Expected																		X	
		Unexpected																		X	

4.2.2 Input from practice on exceptions

Two kinds of practical inputs were gathered: i) empirical data that represent occurred exceptions at TRI pilot, discussed in Section 4.2.2.1, ii) discussion through semi-structured interviews with industry practitioners from the pilots, discussed in Section 4.2.2.2.

4.2.2.1 Empirical data on exceptions from industry

TRI pilot is a representative example of manufacturers operating in high mix-low volume production and dynamic environments. The pilot served as a good base to gather exceptions as occur in practice. The company notes exceptions in a "Rejects Goods" report (shown in Appendix E), which briefly describes information on detection of an exception, causes, risk analysis, and actions to remedy the issues. Information from such reports is stored as data entries in a database source. 1006 data entries were gathered, representing exceptions occurred in a timeframe of two years (February 2017 – February 2019) (Verdonschot, 2020).

The categorization of exceptions resulting from the SLR (Table 11) served as a structured starting point to categorize the exceptions occurred at TRI. Of course, new categories shall be added if are not identified by literature. The analysis of the data is summarized in the categorization of occurred exceptions in practice, shown in Table 12. As can be seen, all occurred exceptions fit the six high-level categories from SLR: resource-related, order-related, process-related, event-related, data-related and various exceptions. At the more concrete levels, new categories have been added, like for instance the work item deviation, referring to exceptions occurring on items that can be repaired, without hindering the continuation of the process or compromising quality levels. Also, more detailed levels have appeared, as for example in the equipment malfunction category in which more detailed types have been identified (e.g., an equipment malfunction can be on the hardware itself, or the software or the method used to operate it). The more detailed types of exceptions are extremely useful for selecting and applying appropriate and efficient handling strategies.

At Table 12, the exceptions fitting under the categories from SLR are highlighted in light-blue. The new added categories that appeared in practice but not in literature are highlighted in light-green. Of course, the sample of 1000 exceptions at a single organization might not be representative to generalize the results, but it gives good insights on how exceptions appear in practice. Regarding the frequency of the exceptions at TRI, work-item deviations appeared the most (29%) as symptoms, with work item failures (27%) and resource malfunction (10%) following. Investigating more thoroughly the symptoms, to find the root cause of an exception, it appeared that a resource malfunction was responsible for nearly half of the occurred exceptions (48% including indirect causes). Given that the machines and equipment at TRI are in general robust, this observation is rather in line with the assumption that the high number of machines might cause issues at production. Moreover, the frequency of exceptions which had an external trigger as a root cause (13%), indicates the dynamicity of environments in which TRI operates, and also the poor integration with external parties.

Table 12: Categorization of (occurred) exception types at TRI pilot. Categories in light-blue match the ones from SLR (Table 11). Categories in light-green appeared in practice but not in literature.

Exception types categories (occurred)				Description/Example
Resource-related	Machine-related	Equipment malfunction	Hardware	Hardware malfunction of a machine during (automated) operation.
			Software	A software-related failure on a machine.
			Method	A malfunction of the method that is used during (automated) operation of a machine.
			Indirect	A malfunction on a machine caused by an indirect action.
			Maintenance	A malfunction caused by insufficient maintenance.
		Machine/equipment breakdown		Exception caused when a machine/equipment stops operating.
	Tool-related	Malfunction		A malfunction on a tool used by a machine or a human operator.
	Material/Product-related	Missing part		Unavailable parts that pause the continuation of a process (e.g., not in inventory).
	Personnel-related	Human error		Any mistake caused by a human action.
Order-related		Deadline expiration		Reaching a deadline for not sending an order to a customer on time.
Process-related		Work item abort/failure		A deviation on work items that cannot be repaired and stop the process. For instance, damaged or deformed items.
		Work item deviation	Direct	A deviation on a work item caused by the process(s) it is involved in. Though, the deviation is repairable (e.g., lose layers that can be tightened) and does not impede the continuation of a process (in contrast to work item abort/failure category).
			Indirect	A deviation on a work item caused by indirect actions outside the scope of the process(s) the item is involved in.
			External	A deviation on a work item caused by an external party.
		Work item unavailable		Work item is missing/not available for further process.
	Time-related	Timeout/Deadline expiration		Reaching a deadline/goal for completing an action/process.
Event-related		External event	Supplier event	Exception caused by a supplier, e.g., delivered items with missing burrs/screws.
			Customer event	Exception caused by a customer.
			Government event	Exceptions caused by government's and/or authorities' changes on regulations and policies.
			Environment/market event	Exceptions caused by environmental circumstances.
Data-related		Ambiguous data		Data that can be interpreted with multiple ways and the selected way leads to errors.
		Incomplete data		Data is not complete to continue a task/process.
		Incorrect data		Data is incorrect that leads to errors.

		Incorrect data (external)		Data coming from external parties (e.g., supplier) is incorrect that leads to errors.
		Over-specified information		Too much (redundant) data that causes errors and hinders efficiency.
		Unknown		Exceptions on data that their type could not be determined.
Various		Software bug		A software-related error on applications.
		Constraint violation	Direct	Inconsistencies in the production process as deviations from the modeled process. For example, non-executed actions that should have been executed according to instructions.
			Indirect	Inconsistencies in an indirect process as deviations from the corresponding modeled process.
			Limitation	The specifications of a product are not achievable/reachable with the specifications of the process in which it is involved.
		Performance analysis		Exceptions arising when delivery performances are not reached by the company or a supplier.
		Unknown		Exceptions whose (root) cause has not been identified.

4.2.2.2 Interviews with industry practitioners on exceptions

More insights on exceptions (and their handling) have been looked for through discussions with industry practitioners. Semi-structured interviews (Adams, 2015) were chosen to allow for better explanations and follow-up questions, compared to static questionnaires. Practitioners from three pilots from the SHOP4CF project were contacted. The two were available for interviews, while the third could only respond on paper. Due to the COVID-19 pandemic measures[52], the interviews were conducted in an online setting.

The structure of the discussion during the interviews has three main parts: questions on the types of exceptions (the high-level categorization from a preliminary SRL was used as a good base), questions on the frequency and criticality of occurred exceptions, and current handling strategies. The structure of the questionnaire is available on Appendix F.1. Of course, as the interviews were semi-structured, the questionnaire was not always followed exactly. Though, important insights were gathered that respond to the main questions. The responses and full transcripts of the discussion are also available in Appendix F.

The outcome of all three interviews is summarized in Table 13. The exceptions fitting under the categories from SLR are highlighted in light-blue. The new added categories that appear in practice but not in literature are highlighted in light-green.

Table 13: Categorization of exception types appearing at SHOP4CF pilots. Categories in light-blue match the ones from SLR (Table 11). Categories in light-green appeared in practice but not in literature.

Exception types categories			Pilot A	Pilot B	Pilot C
Resource-related	Machine-related	Equipment malfunction	Rarely	Rarely	Yes
		Machine/equipment breakdown	Rarely	No	Yes
		Unavailable	Rarely	-	-
	Tool-related	Malfunction	Rarely	No	Yes
	Material /Product-related	Unavailable	-	-	Yes
		Missing parts	Yes	No	No
	Personnel-related	Operator absenteeism/unavailability	-	-	Yes
		Improper operation	Rarely	No	No
		Human error	Yes	-	Yes
Order-related		Change in order (requirements)	Rarely	Yes	Yes
		Cancellation	Yes	Yes	Yes
		Priority change/Rush order	Under conditions	Under conditions	Under conditions
Process-related		Activity failure	No	No	No
		Message-flow failure	Rarely	-	Yes
	Time-related	Timeouts/Deadline expiration	Rarely	-	-
Data-related		Connectivity issue	Yes	-	Yes
Various		Software bug	Rarely	Rarely	Rarely
		Constraint violation	No	-	-

As can be seen, most of the main categories identified in literature (except the event-related category which was not discussed thoroughly during the interviews) appear in practice,

[52] https://ec.europa.eu/info/live-work-travel-eu/coronavirus-response_en

without much frequency though. For example, two pilots indicated that equipment malfunction are rare disrupting events, due to the proper maintenance taken. On the other hand, the third pilot, indicated that while they might haven even daily machine breakdowns/malfunctions, the impact is not very high due to additional/spare production lines (this approach is also discussed in Section 4.3.1.2 as part of handling strategies). Similarly, software bugs are not common issues due to the efficient testing before implementation. However, pilot B mentioned that the first period of deployment of a new system, software errors might appear. Activity failure was not recognized as a category, as the reasons causing an activity to fail have already been identified on other categories. Improper operation (e.g., a part that is fed in a wrong way onto a machine) is not an identified type of exception, however human errors in general are recognized as common errors. Regarding data related exceptions, two pilots indicated that connectivity issues are the culprit for such exceptions.

4.2.3 Designed categorization of exception types

The input from literature (SLR), discussed in Section 4.2.1, and the input from practice, discussed in Section 4.2.2, are synthesized to produce a final categorization of exception types. Categories from SLR (Table 11) are re-structured/revised and combined with outcomes from practice (Table 12 and Table 13) into a single categorization. Design decisions are discussed below. To ease the discussion, the above three categorizations are labeled as follows: C1 for Table 11, C2 for Table 12, and C3 for Table 13. The design decisions are grouped under the self-explanatory categories of *abstraction*, *omission*, *inclusion (enrichment)*, *merge*, *addition* and *move*.

Abstraction

Design decision 1): The final categorization consists of four abstraction levels. The first three are the ones originating from C1 (also appearing in C2 and C3), and the fourth one (the most detailed) originating from C2.

Design decision 2): "Equipment malfunction" and "machine/equipment breakdown" categories from C1 are further detailed into a fourth abstraction level with values from C2. More specifically, "Hardware exception", "Software exception", "Incorrect (use of) method/operation" are added (renamed from C2 for clarity). Category "Indirect" from C2 is not considered as the exception shall be attributed in one of the rest categories (also, that one has very low occurrence frequency which does not justify its inclusion). "Maintenance" from C2 is not considered either, as insufficient maintenance is seen as a root cause of a malfunction or breakdown exception which has to be handled anyways as identified type.

Design decision 3): A material/product can face various defects (Tönnes, 2018), especially when it is a complex one. All types of defects are impossible to consider and not relevant when trying to categorize exception types. Therefore, "Processability issue" and "Quality issue" on third abstraction level of "Material/Product-related" category from C1 is not further detailed.

Design decision 4): "Improper operation" subcategory on third abstraction level under "Personnel-related" category from C1 is renamed into the more generic "Human error", in accordance with C2 and C3. Human errors are common source of exceptions in industry, affected by many factors (Franciosi et al., 2019) (even operator's social life can affect his/her performance (Reyes et al., 2015)). Böllhoff et al. (2016) provide a taxonomy on human errors. However, as this research aims at helping in resolving an exception, it is not of interest to identify the specific cause that led to the human error. It is the result (exception) that finally matters, so actions should be taken based on it (and not on the origin that led to the exception). Moreover, the origin/root cause that has led to a human error, might already have been attributed to a different category of exception. For instance, erroneous or ambiguous information that an operator receives (Calvo Olivares et al., 2018) might well be considered as a data-related exception, which will require corresponding actions. Especially in production scenarios where a task can well be executed both by a robot or an operator, from a process perspective the exception should not be seen as a human error but as a different type of error (e.g., data-related or activity failure).

Design decision 5): The "External events subcategory on third abstraction level under "Event-related" category from C1 are further detailed into a fourth abstraction level with values from C2. More specifically, "Supplier event", "Customer event", "Government event" and "Environment/market event" subcategories are added.

Omission

Design decision 6): Categories labeled as "General" on the third abstraction level of C1 are omitted as they do not provide detailed information that could help in their handling.

Design decision 7): "Task failure (expected/unexpected)" subcategory on third abstraction level under "Machine-related" category from C1 is excluded, as a failure of a machine to perform a task is more relevant from a process perspective (for instance for resource allocation purposes). Consequently, a relevant category is instead placed under "Process-related" category (see Design decision 26).

Design decision 8): "Specification errors" subcategory on third abstraction level under "Machine-related" category from C1 is omitted, assuming that the operating machine/equipment has been selected properly. Moreover, if the specification errors refer to issues using a machine, these are rather covered by the category "Incorrect (use of) method/operation" (see Design decision 2).

Design decision 9): "Delayed delivery" subcategory on third abstraction level under "Material/ Product-related" category from C1 is excluded. The fact that a part has been delivered late means that it was not available when needed. Thus, this type of exception is rather incorporated under the "unavailable" material/product subcategory.

Design decision 10): "Disqualified" subcategory on third abstraction level under "Personnel-related" category from C1 is excluded. The fact that an operator that causes an error is disqualified, is a resource allocation deficiency at first place. In other words, when an error occurs, it has to be solved regardless of being due to insufficient skills of an operator. Similarly, the "training deficiency" subcategory is omitted.

Design decision 11): "Delayed operation" subcategory on third abstraction level under "Personnel-related" category from C1 is excluded. The reason is that delays are covered by time-related exception types from a process perspective, which are also ignorant whether it was a human operator or an automated device that caused the delay.

Design decision 12): "Deadline expiration" subcategory on third abstraction level under "Order-related" category of C2 is not included in the final categorization. Missing a customer delivery deadline means that either delivery activities take longer (which is out of the production scope that this research considers) or production activities has taken longer (which can be due to time-related exceptions from a process perspective).

Design decision 13): "Workflow event" subcategory on third abstraction level under "Event-related" category from C1 are excluded as they rather refer to the general "Process-related" category.

Design decision 14): The "Unspecified events" subcategory on third abstraction level under "Event-related" category from C1 is excluded as it is vague. Also, any event should fit under the rest categories.

Design decision 15): The "Application failure (unexpected data)" subcategory on third abstraction level under "Data-related" category from C1 is excluded as it is not considered purely data-related issue. Application failures are treated as software issues under resource-related categories. Though, the data (any information in general) used during execution of activities, either through a (software) application or verbally is relevant. Issues there can be due to ambiguous data, missing/incomplete or incorrect data, as identified from C2. "Over-specified information" on third abstraction level under "Data-related" category from C2 is not included as an exception type. If an error occurs due to redundant information, it is rather a design issue to help this solved. "Unknown" subcategory on third abstraction level under "Data-related" category is excluded as it is vague.

Design decision 16): "Basic failure" subcategory on third abstraction level under "Various" category from C1 excluded as it is vague.

Design decision 17): "Management problem" subcategory on third abstraction level under "Various" category from C1 is removed as it is rather a general issue on tactical or strategic level and not on operation level that this research considers.

Design decision 18): "Supplier/subcontractor problem" subcategory on third abstraction level under "Various" category from C1 is removed as it is covered by the "Supplier event" subcategory on fourth abstraction level under "External event" category under "Event-related" category (see Design decision 5).

Design decision 19): "Design problem/error" subcategory on third abstraction level under "Various" category from C1 is removed as it typically cannot be resolved within reasonable time frames during production operations.

Design decision 20): "Disqualified working conditions" subcategory on third abstraction level under "Various" category from C1 is removed as this typically refers to continuous situations that lead to such conditions, and not to current disrupting events during production operations.

Design decision 21): "Expected" and "Unexpected" subcategories on third abstraction level under "Various" category from C1 are removed as they do not refer to what is the exception under consideration but on its characteristic whether it is expected or not.

Design decision 22): "Performance analysis" subcategory on third abstraction level under "Various" category from C2 is not included as it is mostly refers to exceptions on tactical or strategic level and not on operation level that this research considers. For example, if a KPI of a process is not achieved, redesign actions are required. Moreover, if it refers to performance on operational level, such issues are mostly covered by the "Quality issue" subcategory.

Design decision 23): "Unknown" subcategory on third abstraction level under "Various" category from C2 is not included as it is vague.

Inclusion (enrichment)

Design decision 24): "Limited operations/Deterioration" subcategory on third abstraction level under "Machine-related" category from C1 is kept in the final categorization to denote exceptions that might not hinder the continuation of the usage/operation of a machine, but actions are needed (e.g., repair, maintenance, change of machine).

Design decision 25): The "Tool-related" category from C1 is enriched with the "Malfunction" subcategory on the third abstraction level, originating from both C2 and C3. The rest three subcategories on the third abstraction level from C1 remain, renaming though the "Occupied" subcategory into "Occupied/unavailable" as the main point is to denote that the tool is not available for use.

Merge

Design decision 26): Process consist of activities (subprocesses or tasks). To avoid ambiguity and duplicates, the "activity" is selected to cover also the terms "operation" and "work item". Thus, the subcategories "Activity failure", "Operation errors", "Work item abort/failure" on third abstraction level of "Process-related" category from C1, together with subcategory "Work item deviation" on third abstraction level of "Process-related" category from C2 are incorporated into the subcategories "Activity abort/failure" and "Activity deviation". The former describes the exceptions that lead to a termination of the activity and alternative paths have to be selected, while the latter describes exceptions where reties are possible to recover the state of the activity and allow for its continuation. Note that the scope of an exception is relevant, i.e., whether it refers to a single task, a subprocess or an entire process. Though, there is no need to further detail the subcategories. Moreover, the term "work item" on the third abstraction level of C2 rather refers to material/product as "items" under work/process. In that respect, their "deviation" and "unavailability" are treated under the "material/product" category. The deviation can be seen either as a processability issue or a quality issue.

Addition

Design decision 27): An "Internal event" subcategory on third abstraction level under "Event-related" category is added to distinguish events happening within the boundaries (physical or functional) of an organization, compared to the external events happening outside the organization. A subcategory “Emergency event” is added on fourth abstraction level under the new “Internal event” subcategory.

Design decision 28): “Connectivity issues” subcategory on third abstraction level under “Data-related” category from C3 rather refers to infrastructure issues. Though, a connection issue typically causes lack of data or wrong data or duplicate data due to synchronization issues. Thus, a subcategory “Data synchronization issues” on third abstraction level under “Data-related” category is added.

Design decision 29): “Software bug” subcategory under “Various” category from C1 is considered an issue on any running software application (e.g., an information system). A software application is deployed on infrastructure (either on premise or on “cloud”), which is seen as resource. Therefore, an “Infrastructure-related” category on second abstraction level is added under “Resource-related” category from C1. Then, an “Application issue” subcategory is added on third abstraction level under that “Infrastructure-related” category. Consequently, the software bug (or any issue) is added on fourth abstraction level under that “Application issue” subcategory.

Design decision 30): “Infrastructure exception” and “Power outage” subcategories on third abstraction level under “Various” category from C1 are both covered under a new subcategory named “Asset issue” on third abstraction level under “Infrastructure-related” category (new category resulted from Design decision 29).

Move

Design decision 31): “Temporal event” subcategory on third abstraction level under “Event-related” category from C1 are moved on third abstraction level of “Time-related” subcategory of “Process-related” category as it refers to time events on activities.

Design decision 32): “Constraint rule” and “Constraint violation” subcategories on third abstraction level under “Various” category from C1 are moved on third abstraction level under “Process-related” category as they refer on constraints on activities. The subcategories “Direct”, “Indirect” and “Limitation” on fourth abstraction level under “Constraint violation” subcategory under “Various” category from C2 are not included as they are deemed as not relevant and sufficient to be generalized.

Design decision 33): “Communication failure” subcategory on third abstraction level under “Various” category from C1 is moved on third abstraction level under “Infrastructure-related” category (new category resulted from Design decision 29).

Based on the above design decisions, the final categorization of exception types is presented in Table 14.

Table 14: Designed categorization of exception types.

Exception types categories			
Resource-related	Machine-related	Equipment malfunction	Hardware exception
			Software exception
			Incorrect (use of) method/operation
		Machine/equipment breakdown	Hardware exception
			Software exception
			Incorrect (use of) method/operation
		Limited operations/Deterioration	
		Unavailable	
	Tool-related	Malfunction	
		Wear	
		Disqualified	
		Occupied/Unavailable	
	Material/ Product-related	Unavailable	
		Missing part	
		Processability issue	
		Quality issue	
	Personnel-related	Operator absenteeism/Unavailability	
		Human error	
	Infrastructure-related	Application issue	(Software bug/issue)
		Communication failure	
		Asset issue	
Order-related		Change in order (requirements)	
		Cancellation	
		Priority change/Rush order	
Process-related		Activity abort/failure	
		Activity deviation	
		Message-flow failure	
		Constraint rule	
		Constraint violation	
	Time-related	Timeout/Deadline expiration	
		Temporal event	
Event-related		Internal event	(Emergency event)
		External event	Supplier event
			Customer event
			Government event
			Environment/market event
Data-related		Ambiguous data	
		Missing/incomplete data	
		Incorrect data	
		Incorrect data (external)	
		Data synchronization issues	

4.3 Operational exception handling

Upon occurrence of an exception, handling mechanisms take place to resolve any issues. Corrective actions typically have the form of handling patterns or strategies, especially for expected exceptions, while for unanticipated exceptions they rely on performing ad hoc interventions during runtime (Marrella et al., 2018). Various strategies and patterns have been identified in literature. However, each strategy is suitable for specific type(s) of exceptions. This research aims to design guidelines on selecting the right strategy/pattern based on the type of exception occurring during operations. The determination of a suitable handling approach also depends on how well the organization is performing or desires to perform. Any KPIs that the organization deploys to measure performance on operation level might affect the selection of handling strategies. For instance, if KPIs on quality are of highest priority for a manufacturer, retaining quality while compromising delivery time is an important factor to select corrective actions when exceptions occur.

This section presents the design of operational exception handling guidelines. First, exception handling strategies/patterns are studied in Section 4.3.1. KPIs for MOM operations are considered in Section 4.3.2. The design of the guidelines is described in Section 4.3.3.

4.3.1 Existing exception handling approaches

This section identifies existing exception handling approaches. Literature is the main source of knowledge, as discussed in Section. 4.3.1.1. Practice has also been consulted to check whether missing or non-identified strategies/patterns exist, as discussed in Section 4.3.1.2.

4.3.1.1 Exception handling approaches from literature

The SLR presented in Section 4.2.1 covers, apart from exception types, handling approaches as well. This has been taken into consideration in the search terms. The resulting short list of studies provided a solid base to look for exception handling strategies or patterns. Table 15 lists the most prominent approaches to handle exceptions. The approaches are named in a rather straight-forward way (further explanation is given later when needed). Obviously, common approaches (with same or similar labeling) have been identified by many researchers. The list is revised in Section 4.3.3.1 to avoid duplicates/overlaps and to create a final list of exception handling approaches.

Keddis et al. (2016) present only a few options, but with an explicit focus on manufacturing. De Snoo et al. (2010) mostly focus on production planning approaches. A process perspective is considered in many studies (e.g., Reichert & Weber (2012b), Russell et al., (2006a) or the work of Mourão & Antunes (2005), Reichert & Weber (2012a) on unexpected exceptions), without, though, (clear) reference to manufacturing. While their identified approaches are relevant and valid, their applicability in the physical world of manufacturing needs investigation, especially for exception types refereeing to physical objects (e.g., machine breakdown). It is important, thus, to provide some guidance on which handling approach to select based on the type of occurred exception.

Table 15: Exception handling strategies/patterns as appear in literature.

Exception handling strategies/patterns	Study
• Retry assembly • Discard workpiece • Replace order	Keddis et al. (2016)
• Compensation activity • Retry activity • Continue process • End process and propagate exception • Terminate process	Ritter & Sosulski (2014)

• Adding or deleting process fragments
  o Insert process fragment
  o Delete process fragment
• Moving or replacing process fragments
  o Move process fragment
  o Replace process fragment
  o Swap process fragment
  o Copy process fragment
• Adding or removing process levels
  o Extract subprocess
  o Inline subprocess
• Adapting control dependencies
  o Embed process fragment in loop
  o Parallelize process fragments
  o Embed process fragment in conditional branch
  o Add control dependency
• Change transition conditions
  o Update condition
	Reichert & Weber (2012a)

• Trying alternatives
  o Ordered alternatives
  o Unordered alternatives
• Adding behavior
  o Immediate fixing
  o Deferred fixing
  o Retry
  o Rework
• Cancelling behavior
  o Reject
  o Compensate
• Resource-related handling patterns
  o Delegation
  o Escalation
  o Reallocation (stateful/stateless)
  o Deallocation
• Flexible handling
  o Suspension/Resumption
  o Skipping
  o Redo
  o Pre-do
  o Cancel
	Reichert & Weber (2012b)

• Allocate damage • Repair (production) plan • Replan • Improve plan • Refer upwards or start lateral coordination • Do nothing	De Snoo et al. (2010)
• Trying alternatives o Ordered alternatives	Lerner et al. (2010)

o Unordered alternatives • Adding behavior o Immediate fixing o Deferred fixing o Retry o Rework • Cancelling behavior o Reject o Compensates	
• Taking no measures • Rollback operation • Skipping exception node • Compensation operation	Wu (2009)
• Remove work item • Remove case • Remove all cases • Suspend work item • Suspend case • Suspend all cases • Continue work item • Continue case • Continue all cases • Restart work item • Force complete work item • Force fail work item • Compensate	Adams et al. (2007)
• Work item level handling o Continue offer o Reoffer o Force-fail-offer o Force-complete-offer o Continue-allocation o Reallocate o Reoffer-allocation o Force-fail-allocation o Force-complete-allocation o Continue execution o Restart o Reallocate-start o Reoffer-start o Force-fail o Force-complete • Case level handling o Continue workflow case o Remove current case o Remove all cases • Recovery action o No action o Rollback o Compensate	Russel et al. (2006a)
• Abort o Hard o Compensate some tasks o Compensate all tasks • Decrease completion time to meet deadline • Recover from a system failure condition and replace the system in automatic mode • Recover from a task failure and place the system back in automatic mode	Mourão & Antunes (2005)

• Recover to achieve the lowest penalty possible, i.e., to minimize the impact • Jump forward to a task in the work model • Repeat a previous task that was not executed in the desired way • Jump backwards in the work model and compensate some already executed tasks • Delay this task • React to environmental changes	
Masking ○ Hierarchical ○ Group • Workflow recovery ○ Ignore ○ Warning ○ Retry ○ Suspend/Stop/Resume ○ Backward recovery ○ Forward recovery • Workflow changes ○ Modifying Justified Event-Condition-Action (JECA) rules ○ Inserting JECA rules ○ Removing JECA rules ○ Commutative change ○ Combination of the above • Case-based reasoning ○ Retrieval of the most similar cases to the identified exceptional situation ○ Analysis of the solution from the most similar cases ○ Adaptation of the most similar cases ○ Updating of the system by adding the verified solution to the case repository	Luo et al. (2000)

4.3.1.2 Exception handling approaches from practice

With the list of exception handling approaches from literature as a solid base, input from practice is considered to check whether other approaches are applied. The empirical data of TRI pilot used for exception types (Section 4.2.2.1) is a good source. Moreover, the discussion with industry practitioners (introduced in 4.2.2.2) is additional useful input.

Analyzing the TRI datasource, with around 1000 data entries, many of the literature approaches are encountered in practice, as shown in Table 16. Extra categories have been identified, which are not relevant for further analysis (“Not operation related”, “Unknown”).

Table 16: Exception handling actions taken at TRI pilot.

Exception handling actions	Frequency	Supported by literature
• Abort	453	Keddis et al. (2016) Reichert & Weber (2012) Adams et al. (2007) Russel et al. (2006a) Mourão & Antunes (2005)
• Meet deadline	2	Mourão & Antunes (2005)
• Recover from system failure	1	Mourão & Antunes (2005)
• Minimize impact	1	Mourão & Antunes (2005)

• Repeat previous task	20	Keddis et al. (2016) Ritter & Sosulski (2014) Reichert & Weber (2012) Adams et al. (2007) Russel et al. (2006a) Mourão & Antunes (2005)
• Jump backward	68	Mourão & Antunes (2005)
• Delay task	37	Mourão & Antunes (2005)
• Compensate some tasks	59	Ritter & Sosulski (2014) Reichert & Weber (2012) Wu (2009) Adams et al. (2007) Russel et al. (2006a) Mourão & Antunes (2005)
• Contact internally	15	Ritter & Sosulski (2014) Reichert & Weber (2012) De Snoo et al. (2010)
• React to environmental changes	12	Mourão & Antunes (2005)
• No action	48	Ritter & Sosulski (2014) Reichert & Weber (2012) De Snoo et al. (2010) Wu (2009) Russel et al. (2006a)
• Not operation related	138	-
• Unclear	120	-

The discussion with practitioners (transcripts available in Appendix F) did not reveal any new handling approaches. All three pilots apply regular approaches as appear in literature. Pilot C raised an interesting point with respect to handling equipment-related exceptions. They mentioned that in a few places in their production facilities they have additional production lines that take over production. While this is definitely an approach to avoid downtimes, it requires in advance design/configuration of facilities. From an operational point of view (scope of this research on exceptions), that approach cannot be generalized given that not every manufacturer can provide/support this alternative. It will be included, though, as an option in the selected list of exception handling approaches (revised in Section 4.3.3.1).

4.3.2 MOM KPIs

Key Performance Indicators (KPIs) represent a set of critical measures of organizational performance for current and future success (Parmenter, 2010). KPIs quantitatively evaluate the performance of organizations and are driving forces towards their strategic objectives (Wang & He, 2012). KPIs are implemented at multiple levels in an organization, but the focus of the current research is on the ones at MOM level. Typical performance metrics on operational level are Overall Equipment Effectiveness (OEE), as a multiplier of Availability × Performance × Quality of equipment, and First Time Yield (FTY) (also labeled as First Pass Yield), as percentage of products that are manufactured correctly (according to specifications) the first time through the manufacturing process without scrap or rework.

Developing performance metrics vary per industry and organization (Muhammad et al., 2018). Furthermore, manufacturing enterprises might implement just a few of a long list of available measurements. The implemented KPIs can have their own criticality based on organizations strategies, which means priorities on getting/keeping high values on some metrics over others is often the case. When exceptions occur, the values of the implemented KPIs have to be taken into account for choosing optimal corrective actions. For instance, in case of a product defect, a low score on a KPI measuring quality (which is deemed with high priority) will guide operators on retaining/achieving quality targets instead of meeting delivery deadlines.

In order for this research to use well-defined and widely accepted KPIs in the operational exception handling guidelines, literature is consulted. The MESA organization performed a survey among manufacturers and associations to help identify the most important performance metrics. The survey resulted in 28 metrics (MESA, 2014) being used the most by manufacturers. These are grouped in eight categories based on the associated top-level area of improvement[53]:

- Improving customer experience and responsiveness,
- Improving quality,
- Improving efficiency,
- Reducing inventory,
- Ensuring compliance,
- Reducing maintenance,
- Increasing flexibility and innovation,
- Reducing costs and Increasing profitability.

As can be seen from the categories, there are KPIs covering a broader spectrum than production, like for example the "Net Operating Profit" (measuring the financial profitability for all investors/shareholders/debt holders for a manufacturing plant or business unit) under last category.

ISO created the 22400 standard "Automation systems and integration — Key performance indicators (KPIs) for manufacturing operations management", consisting of two parts: "ISO 22400-1: Overview, concepts and terminology" (International Standards Organization, 2014a) and "ISO 22400-2: Definitions and descriptions" (International Standards Organization, 2014b). The standard includes a list of 34 KPIs with a clear focus on MOM level. Apart from being a standard to be used in industry, it has also received academic interest (Kang et al., 2016; Muhammad et al., 2018; Zhu et al., 2018). Of course, the list of the 34 metrics cannot be directly applicable in all manufacturing scenarios, but improvements and corrections are needed (Zhu et al., 2018). To give some structure and refinement in the 34 ISO 22400 KPIs, Kang et al. (2016) provide a categorization, illustrated in Figure 75. They define metrics as "Supporting elements", "Basic'KPIs" and "Comprehensive KPIs".

---

[53] "28 Manufacturing Metrics that Actually Matter (The Ones We Rely On)" (Mark Davidson) - https://blog.lnsresearch.com/blog/bid/188295/28-manufacturing-metrics-that-actually-matter-the-ones-we-rely-on

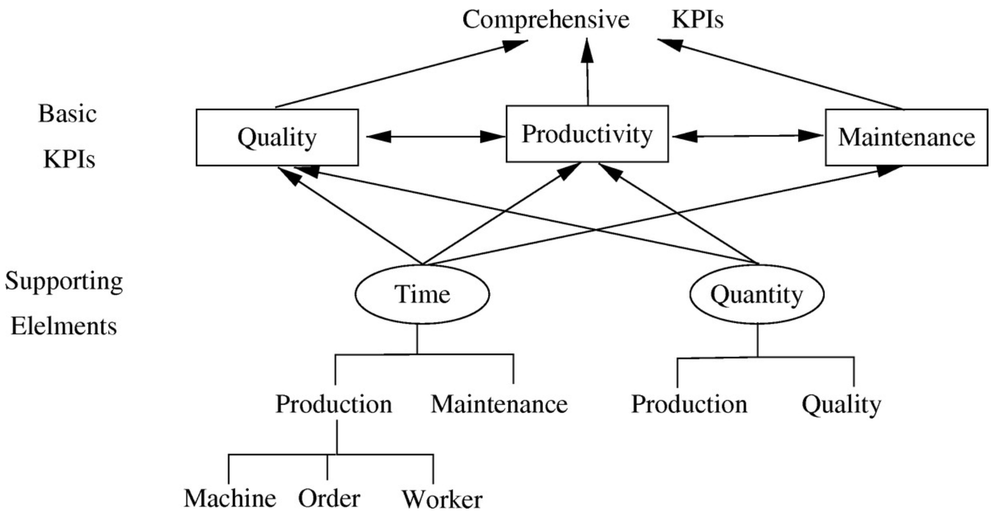


Figure 75: Categorization of MOM KPIs (Kang et al., 2016).

For the purposes of this research, it is not needed to provide detailed definitions of either the 28 MESA or 34 ISO 22400 standard KPIs. Especially when each organization can modify these metrics per need. It is the general categories that give an indication for which KPI to consider during the exception handling process. Given also that the scope is on operational processes, the following categories of KPIs are further considered:

1. Customer experience/Delivery time
2. (Product) Quality
3. Efficiency/Productivity
4. (Production) Costs

4.3.3 Designed operational exception handling guidelines

This section presents the artefact of the operation exception handing guidelines and their design. First, the long list of exception handling strategies/patterns identified in Section 4.3.1 is revised to keep a compact and unique list to be used in the design. Section 4.3.3.2 discusses the steps that lead to the designed guidelines.

4.3.3.1 Selected exception handling approaches

Taking the process perspective as the leading one, instead of resources or data for example, the main approaches to handle an exception on operation level can be grouped in eight categories:

1. Retry, i.e., repeating an activity.
2. Try alternatives, i.e., perform different actions and/or using different equipment to achieve the objective.
3. Compensate, i.e., perform some activities to bring the system back to same state before the exception occurred.
4. Rollback, i.e., undo some activities to bring the system in a previous state.

5. Suspend/Resume, i.e., pause an activity until exception is resolved and resume it afterwards.
6. Continue process, i.e., move to the next activities.
7. Terminate, i.e., stop an activity/process.
8. Escalate, i.e., look for support at different resources.

These eight categories consolidate the approaches of literature and practice. Each of the category represents a direction towards corrective solutions, including a few more detailed approaches. The final list of the selected exception handling approaches is summarized in Table 17.

Table 17: Identified exception handling approaches after literature and practice consolidation.

Exception handling strategies/patterns	
1. Retry	a. Restart
	b. Rework
	c. Immediate fixing
	d. Deferred fixing
2. Try alternatives	a. Replace order
	b. Change/Move to new settings
3. Compensate	a. Add/Insert tasks
4. Rollback	a. Jump backwards
5. Suspend/Resume	a. Delay the activity
	b. Suspend/Resume case
	c. Suspend/Resume all cases
6. Continue process	a. Skip
	b. Jump forward
	c. Ignore
	d. Do nothing
7. Terminate	a. Force complete activity
	b. Force complete case/process
	c. Force fail activity
	d. Force fail case/process
	e. Discard workpiece
8. Escalate	a. Reallocate
	b. Terminate process (7) and propagate
	c. Contact/Coordinate

Of course, the approaches above are not always applied exclusively to each other but can be combined in a stepwise fashion.

4.3.3.2 Design of operational exception handling guidelines

The main objective is to determine a suitable exception handling approach when an exception occurs on operational level. It is assumed that the type of exception has already been determined (note that the detection of an exception is out of scope as has already been discussed) and it is not relevant for the purposes of this research weather the issue is considered a symptom or the root cause. Moreover, at a conceptual level, it is not relevant if the guidelines are performed by a human operator (manually) or by a software system (automatically). The operationalization of the guidelines, discussed in Chapter 6, further details who can perform the step/decisions.

The operational exception handling guidelines consist of a set of decision trees, each of which leads, through logical steps, to the determination of a suitable exception handling approach. For each of the exception types categories, considering also the MOM KPIs categories, a corresponding decision tree is designed. Considering the 9 categories on first/second abstraction level of Table 14 and the selected 4 KPIs categories, 36 decision trees can be designed. However, as for some exception types the determination of a handling approach can be similar, some decision trees are identical, i.e., the same guidance is applied for two or more exception types categories. Similarly, for some exception types the importance of a KPIs is not relevant - in other words, the selected handling approach is not affected by a critical KPI. Thus, the set of identical decision trees is smaller than the maximum product of exception type × KPI category.

More specifically, the KPI category does not differentiate the exception handling selection process (questions/steps in the decision trees) between machine-related and tool-related exception types, as both machines and tools are of same nature or general usability in a production process. This is not the case for material/product related exceptions. Material/products are more consumable (different life cycle) than machines/tools and exist in larger quantities. Regarding the effect of the KPI category on material/product exceptions, think of the following example: when there is a material processability issue, a quality KPI that is of higher priority compared to a time KPI will probably lead to a "retry" or "discard item" handling approach instead of a "continue process to not waste time" approach. On the contrary, the KPI category does not affect the exception handling selection process for infrastructure or data related exceptions. Infrastructure is rudimental whatever the main objectives of an organization are. This also holds true for data issues, which have to be solved as soon as possible, especially in smart and digital manufacturing environments where data is dominant. Regarding order-related exceptions, as they mostly relate to time aspects or requirements, quality KPIs do not have any influence, assuming that the quality has to be preserved the same. Thus, no decision tree is required in that combination.

Consequently, there have been designed 25 unique decision trees based on exception types categories and MOM KPIs categories, labeled in the matrix of Table 18.

Table 18: Matrix of designed decision trees (coded) per exception type category and MOM KPIs category

Exception types categories		**MPM KPIs categories**			
		Time	**Quality**	**Efficiency**	**Costs**
Resource-related	Machine-related	TMT	QMT	EMT	CMT
	Tool-related				
	Material/ Product-related	TMP	QMP	EMP	CMP
	Personnel-related	TPE	QPE	EPE	CPE
	Infrastructure-related	INF			
Order-related		TOR	-	EOR	COR
Process-related		TPR	QPR	EPR	CPR
Event-related		TEV	QEV	EEV	CEV
Data-related		DAT			

For the interest of brevity, only two decision threes are shown here, with the rest available in Appendix G. Namely, the ones corresponding to process-related exceptions for (delivery) time and (product) quality KPIs, TPR and QPR accordingly. The legend used in the decision trees is explained in Figure 76.

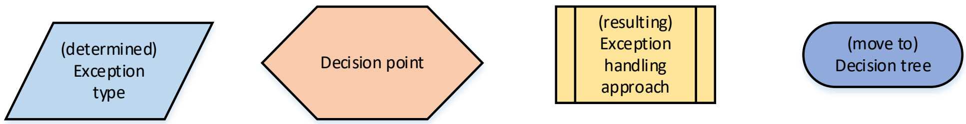


Figure 76: Legend for symbols used in decision trees for selecting suitable exception handling approach.

TPR and QPR are shown in Figure 77 and Figure 78 respectively. As can be seen, there are of course common decision points/paths between the two decision trees but the leading KPIs provide differences. For instance, in case time is more crucial that quality, deferred fixing of a work item is not an option.

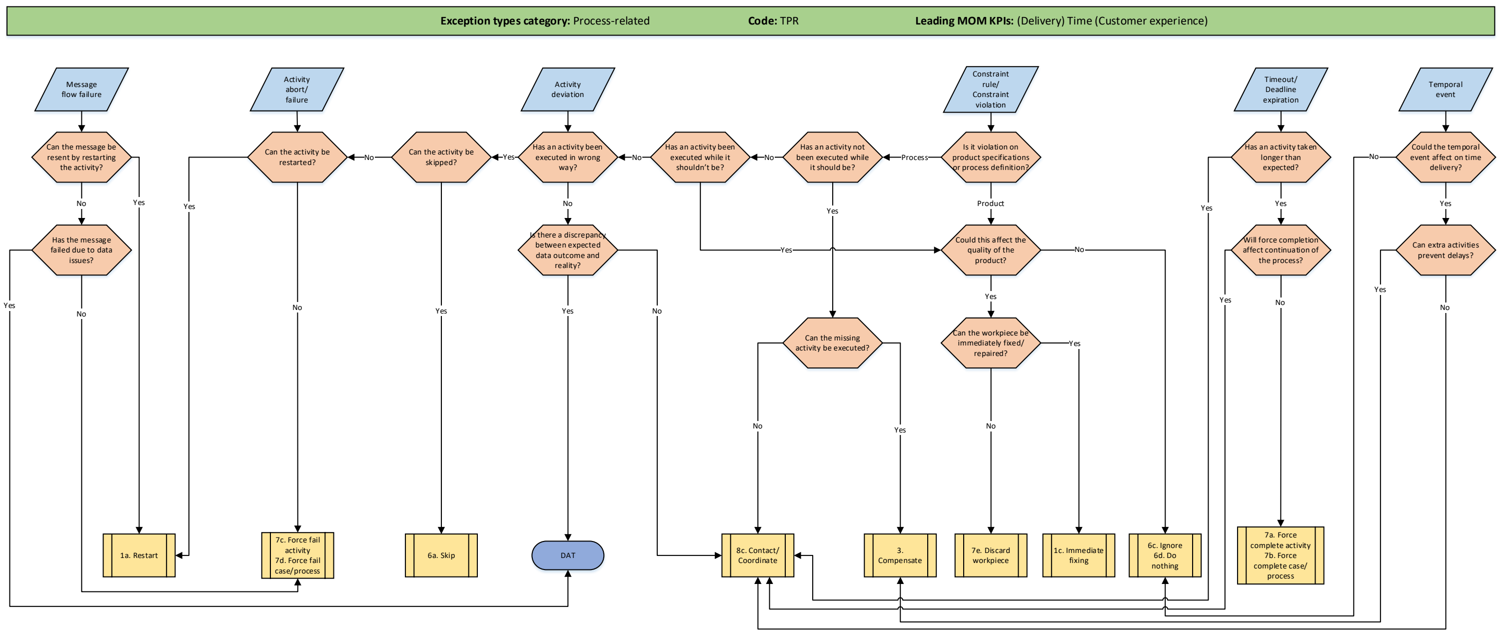


Figure 77: Decision tree for Process-related exception types categories with (Delivery) Time as the leading MOM KPIs.

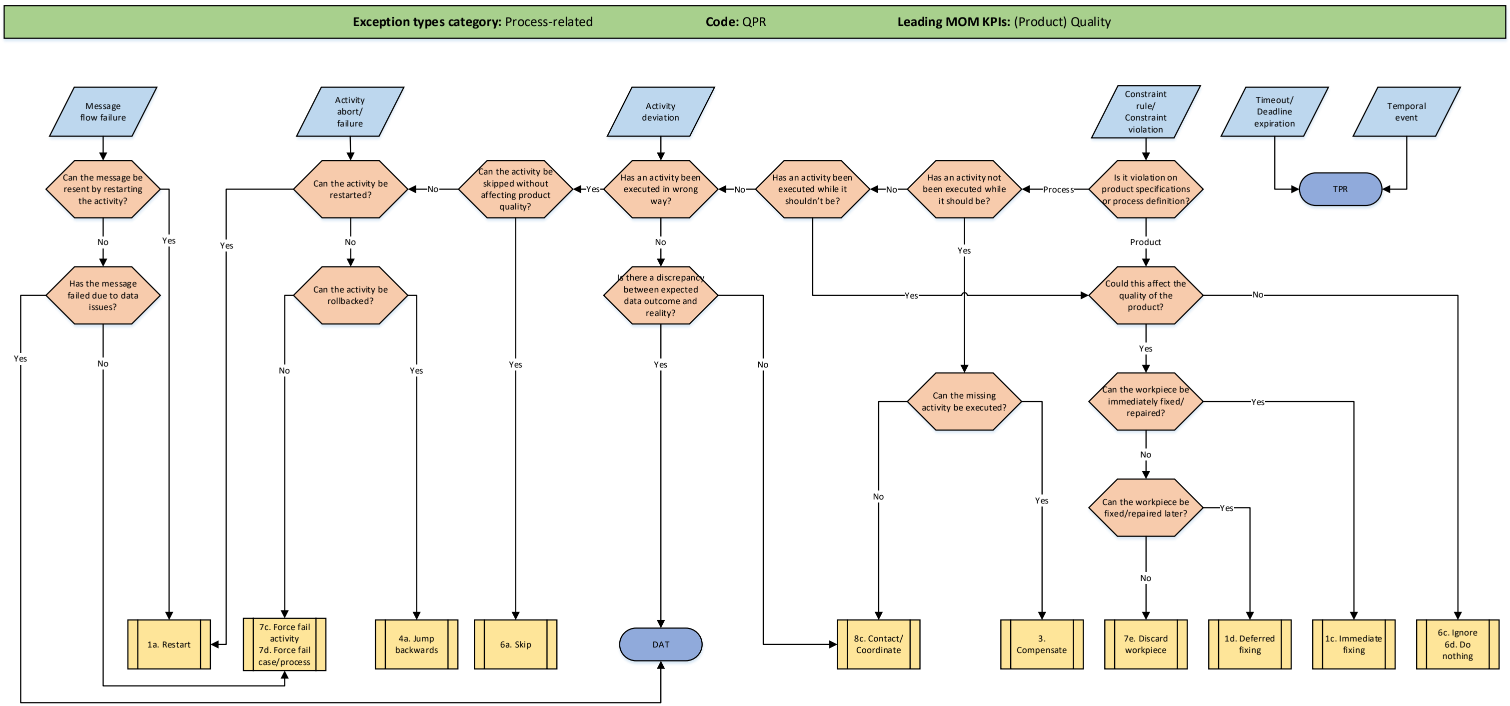


Figure 78: Decision tree for Process-related exception types categories with (Product) Quality as the leading MOM KPIs.

4.4 Chapter conclusion

Given the rise of exceptions in dynamic manufacturing environments, with a clear effect on complexity, it is essential to provide guidance on handling those, with a scope on operational level. Thus, this chapter tackles the topic of exception handling in smart manufacturing, by designing two artefacts: 1) a categorization of exception types, 2) a set of guidelines, in the form of decision trees, to select a suitable exception handling approach.

The categorization of exception types is a result of a DSR design process, in which exception types from literature, through an SLR, are consolidated with exceptions occurring in practice, through interviews with practitioners. Apart from the exception types, MOM KPIs play a role in selecting the right approach to handle any deviation behavior. These two inputs, the categories of exception types and the categories of MOM KPIs, lead to a set of decision trees that through logical paths result in suitable handling approaches.

A preliminary version of the two artefacts had been evaluated with practitioners from three manufacturing companies to get an impression on their utility. The evaluations had been performed in the frames of a master thesis project (Verdonschot, 2020), which the author of the current thesis was guiding and supervising. The operational exception handling guidelines were applied in specific scenarios encountered in the companies. Practitioners were asked to give their opinion on perceived ease of use (PEoU) and perceived usefulness (PU) (Davis, 1989). The positivity on intention to use gave confidence that the design is on the right direction. A main conclusion was that the methodology requires specialization per scenario to be applicable, which was rather expected as its goal is to provide a generic approach. Another interesting insight is that the type of organizational structure (Lunenburg, 2012) might affect its applicability. For instance, in environments with less structure, where operators have a lot of flexibility and freedom to perform actions, the guidelines might be less efficient to use than exploiting operators' expertise and experience.

The conceptual artefacts of the current chapter are operationalized through the advanced MPMS, presented in Chapter 6.

CHAPTER 5

Process automation and integration

Automation of production tasks has well been achieved since the previous industrial revolutions. All recent technological developments in robotics, though, pose challenges in how these fit into existing automation approaches (Monostori, 2014). Robotic solutions are often employed in disparate work cells, following a vertical orientation in their robot control processes. Given also that different technologies are controlled by different systems, it is challenging to provide horizontal, cross-functional process integration that Industry 4.0 demands (Brettel et al., 2011; Kagermann et al., 2013). Existing infrastructures are not ready to support such integration (da Xu et al., 2018).

A demand chain manufacturing setting requires real-time process integration in the entire chain of activities, i.e., from order reception till delivery. While poor process integration can be attributed to many reasons (as discussed in Section 2.1.6.3), in the context of enterprise information processing the integration is essential to guarantee that the right business and manufacturing activities are performed at the right time by the right actors with the right information. This is the technological domain of a business process management system, which is often employed to achieve or improve enterprise integration (van der Aalst, 2013). Such a system can orchestrate manufacturing processes within the scope of a shop floor, or within the entire enterprise by including also business processes, or even at the manufacturing network level across several enterprises (Mehandjiev & Grefen, 2010). A BPMS that is used in manufacturing is also referred to as Manufacturing Process Management System (MPMS).

Thus, the purpose of this chapter is to provide an answer on how end-to-end process integration can be enabled (RQ3) by designing a process management information system for manufacturing operations.

5.1 Chapter outline

As this research advocates the use of BPM theories and technologies in smart manufacturing, it is prudent to first justify why a single process management system for manufacturing operations is of interest (as Gregor & Hevner (2013) argue that any research following the DSR paradigm should be *interesting* and *non-trivial*). This is discussed in Section 5.2. Then, Section 5.3 presents the conceptual design of MPMS, following a well-defined design process and based on design principles. As the system is based on BPM technologies, its design is inspired by existing reference architectures for BPMS and their origin workflow management systems (WFMS). Section 5.4 views MPMS in the context of a CPS, by specifying the required interfaces to other information systems for achieving integration. The conceptual design of the internal functionality of MPMS, enhanced with the interfacing capabilities compose the specification of the process management system to enable process automation and integration. The above structure of the presented contents is illustrated in Figure 79, following the DSR approach (which zooms-in the aspect under concern of the three-aspects design and development of Figure 13). The chapter concludes in Section 5.5.

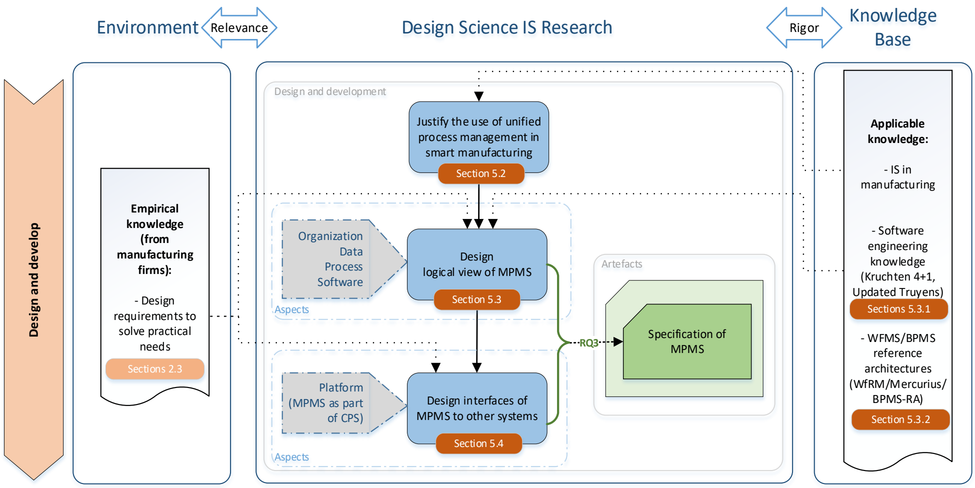


Figure 79: DSR approach for RQ3 - Design of a Manufacturing Process Management System for end-to-end process management.

5.2 Process management support by information systems in manufacturing

As has already been explored in Section 2.1.4, various information systems exist to support the functions of the top levels of the functional hierarchy control (Figure 21). An ERP system covers Level 4 functions, an MES covers Level 3 functions, and a PLC system covers Level 2 functions. Each of these systems offers process management functionality but the support is isolated within their respective functional boundaries (as discussed in Section 2.1.6.3). An ERP system can coordinate business activities but does not have an overview picture of manufacturing operations, which is typically provided by an MES. Of course, this separation is important but, as Industry 4.0 demands for horizontal and vertical integration, cross-functional process management is necessary.

The orchestration of activities across different functional levels requires well-integrated and interoperable systems. The integration of traditional enterprise information systems is ongoing research (Avvaru et al., 2020; Boiko et al., 2020), while interoperability issues (mostly on data aspects) still exist (Pereira et al., 2020). Moreover, cross-functional process management is often hindered by the different modeling approaches and notations to represent processes in different levels (for instance, BPMN used for Level 4 processes and VSM for Level 3). While there exist approaches to use common modeling languages (e.g., use of BPMN for both Level 4 and Level 3 processes by Prades et al. (2013)), or to enable transformation of one modeling style to another (e.g., transformation of Level 4 BPMN models into Level 3 functional charts by Gerber et al. (2014)), a more holistic approach for process management is needed that covers not only process modeling but also execution support.

This research proposes a single process management system to model and enact business and manufacturing activities. This system can unify all the different process management functionality offered by traditional systems across Levels 4 and 3 of the functional hierarchy and interface directly with systems in Level 2. To do so, such a system is positioned as an infrastructure component that supports the functionality of various information systems. The case of a centralized manufacturing process management system has been proposed in (Erasmus, Vanderfeesten, Traganos, & Grefen, 2018) and illustrated in Figure 80. A typical communication flow is the following: MPMS receives order information from an ERP system and detailed production scheduling from an MES. It then initiates the execution of activities by sending work instructions to operators through human interfaces and/or to automated agents through their control systems. Accordingly, MPMS receives progress status on the activities and the state of the agents, informing back also the ERP system on order updates. The communication across systems is typically realized through middleware systems, which are omitted in Figure 80 for sake of simplicity.

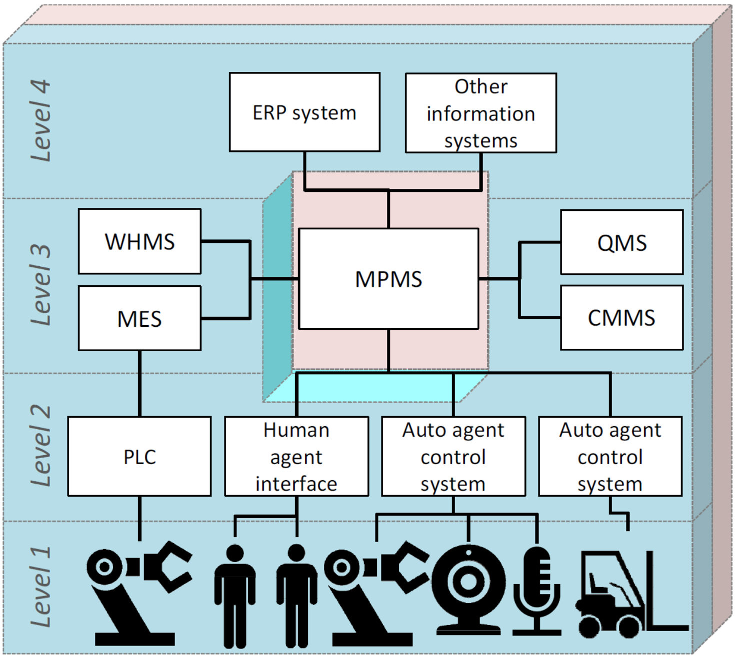


Figure 80: MPMS as an orchestration hub for process management across various information systems on level 2, 3, and 4 of the functional hierarchy. The blue-highlighted (front) horizontal layer represents the application layer, while the pink-highlighted (back) layer represents the infrastructure layer (Erasmus, Vanderfeesten, Traganos, & Grefen, 2018).

It should be noted that MPMS does not replace existing systems, especially those in Level 3, but rather shows the essence of a central process management hub to integrate functions across levels. Especially for SMEs that do not obtain an MES system, MPMS can play the integrative and orchestrating role. For those enterprises that do use an MES system, MES can still interact with factory floor systems (as shown with the direct interface of MES and PLC in Figure 80), but MPMS can take over the process management functionality. This of course depends on the level of process support that commercial MES systems offer and the level of control that each of MES and MPMS should have (Kandasamy, 2021).

5.3 Architecture of MPMS

This section presents the design of the MPMS, by specifying what components and functionality the single process management system proposed in Section 5.2 should entail. The architecture of the system follows a well-defined design process and design principles, discussed in Section 5.3.1. The design is inspired by existing WFMS/BPMS reference architectures, which are briefly introduced in Section 5.3.2. The design is then described in Section 5.3.3.

5.3.1 Design process and design principles

The design of an information system can be a complex process. Proper separation of concerns is required to be able to limit attention of this process in the right context to the right elements and aspects (Garcia et al., 2004; Moreira et al., 2005). In other words, separation of concerns avoids taking everything into consideration at every step of the way.

Kruchten (1995) has defined an architecture framework (abbreviated in this thesis as K4+1) that has become one of the most important standards in thinking about structuring software architectures and guiding their design processes. The main idea of the framework is to have a separation of concerns with respect to phases of the architecture specification and software realization process. The framework organizes the description of an architecture around four main views with their respective main stakeholders:

1. The logical view is concerned with the functionality of the system, by specifying the logical structure of the application with abstract components and their relationships. The prime stakeholders are the end-users of the designed system.
2. The development view deals with the software development management. It specifies the software realization of the system, based on its logical view. Software engineers (programmers) are the main stakeholders.
3. The process view is concerned with the behavior aspects of the system design, i.e., how the components or modules in the logical view collaborate concurrently. Typically, integrators of the system are related with this view.
4. The physical view specifies the hardware on which the software is deployed. System engineers are responsible for the system installation and maintenance.

The differences in the above four views, both in their prime concerns and the main stakeholders involved, may result in a divergence of understanding how the system should be designed and work. To avoid this, a fifth element needs to converge, content-wise, the four views:

5. The scenarios illustrate the four basic views in the form of use cases. They give concreteness and a clear description of the system's functionality, so the associated stakeholders of the four views have a common understanding during the design process.

The five elements lead to the K4+1 model, shown in Figure 81.

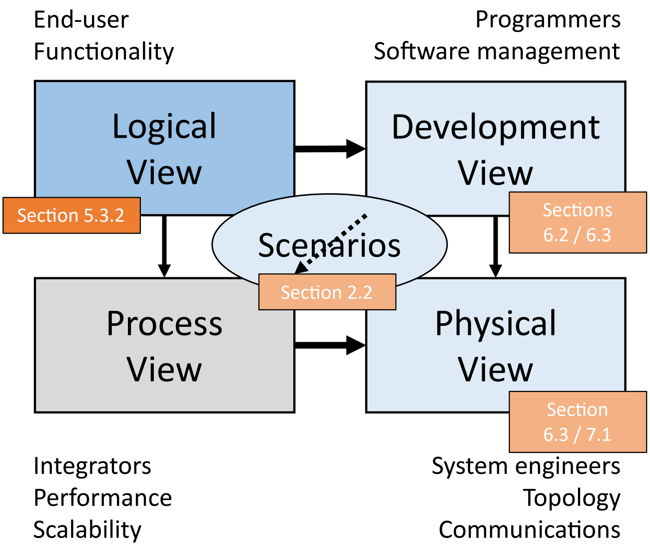


Figure 81: Kruchten (K4+1) framework (Kruchten, 1995) to sequence the development process of MPMS.

The development process starts with the logical architecture, i.e., the specification of the functional structure in abstract terms (no specific implementation or deployment details provided), discussed in Section 5.3.2. The inputs for the conceptual design are the requirements listed in Section 2.3 and the scenarios presented in Section 2.2. The actual software implementation of the logical architecture is covered in the development view. For the scientific interest of this research, not all technical details are presented in this thesis, but only the main developments that realize the conceptual developments (e.g., realization of the activities synchronization mechanism presented in Section 3.5). The software deployment takes places within the physical view of the framework. As the focus of the current chapter is the conceptual design of the system, only the logical view is discussed in the rest of the chapter, while the development and physical views are discussed in Chapters 6 and 7. The process view, which covers the operational integration of the developed system (e.g., software testing, performance improvements, etc.), is of less importance from a scientific perspective (it mostly relates to good software engineering practices) and left out of discussion.

The K4+1 framework provides a separation of concerns in terms of software development phases but does not separate various aspects of the description of a complex software or information system. Therefore, K4+1 is complemented with an updated version of the 5-aspect framework of Truijens (abbreviated in this thesis as UT5). That framework was originally developed for information system development in the '90s (Truijens, 1990) and thereafter updated for information system development in a modern, networked context (Grefen, 2016). The UT5 framework consists of five interconnected architecture aspects, illustrated in Figure 82 and described below (Grefen, 2016).

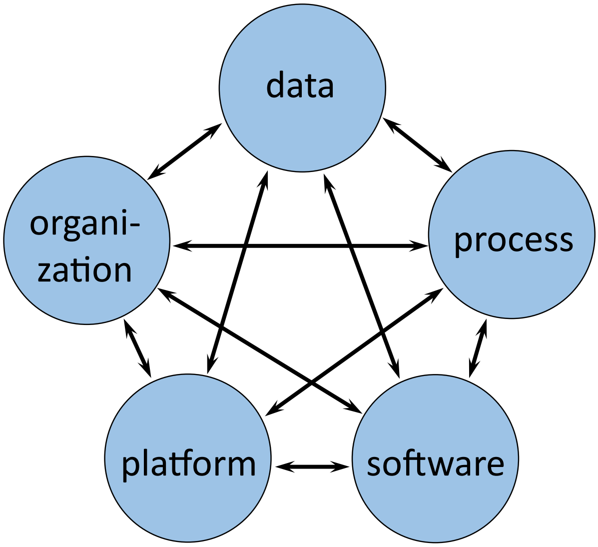


Figure 82: Updated 5-aspect Truijens framework (UT5) (from (Grefen, 2016)) for separating the specification of the architecture of MPMS.

- The software aspect describes the structure of the software under development; it can take the form of UML[54] component diagrams.
- The process aspect describes the structure of business/manufacturing processes that the system supports; it is described for instance in UML activity diagrams or BPMN models.
- The data aspect describes the structure of data manipulated by the system, as well as the structure of the concepts that underlie data definitions (concept model); it is described for instance in UML class diagrams.
- The organization aspect describes the structure of stakeholders in the system's context, such as developers or end-users; it is described by organigrams and/or actor models.
- The platform aspect describes the structure of the technology (both hardware and software) that is required to run the system under design in an operational form.

General information systems and software design approaches are important when developing a system like MPMS, but as the system is meant to support Industry 4.0 endeavors, it should also adhere to contemporary paradigm's principles. The RAMI 4.0 framework (shown in Figure 23) is selected as a relevant and generally accepted one to derive such principles.

With respect to the *life cycle & value stream* dimension, there should be a clear distinction between the type and instance of a product and its processes (DIN/DKE, 2016). A product (any physical part in general) is being designed/specified once and is typically produced in multiple instances (unless it is a unique product, which even in that case there is a clear distinction between design and production). Similarly, processes and activities required to manufacture products are designed/modeled/specified and then multiple instances of those are enacted.

[54] https://www.andrew.cmu.edu/course/90-754/umlucdfaq.html

With respect to the *hierarchy levels* dimension, there should be a clear distinction on the scope of control that MPMS provides. RAMI 4.0 references the physical hierarchy of IEC62264-1 standard (shown in Figure 18) to establish a naming convention for sections and location of manufacturing enterprises and their factories. In that hierarchy, MPMS should provide global control of process in the context of a production area or production line (across multiple work cells), while local control within the context of individual work cells should be responsibility of other systems.

The two aforementioned design principles lead to separation of concerns into two dimensions, illustrated in Figure 83.

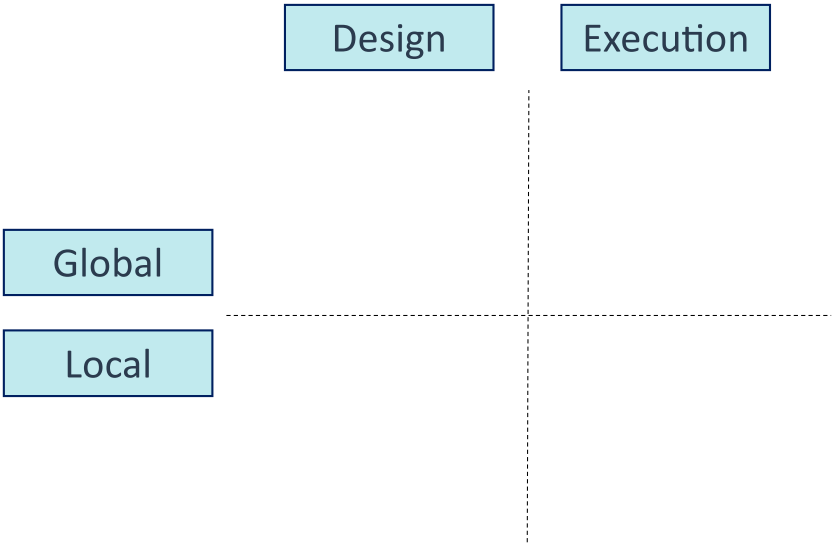


Figure 83: Separation of concerns with respect to life cycle & value stream and hierarchy levels dimensions of RAMI 4.0 framework.

With respect to the *layers* dimension of the RAMI 4.0 framework, there should be a distinction on representation of entities and assets from various architecture perspectives. For instance, a production line can be viewed from a business perspective (e.g., goals to achieve), functional perspective (e.g., workflow of activities), or asset perspective (e.g., set of machines). Apart from the levels of the physical hierarchy, the architecture axis of the framework can be applied on entities such as processes, activities and actors. These entities should be separated between their physical substance (e.g., location of executing an activity, size of materials involved, etc.) and their functional perspective (e.g., purpose of executing an activity). The various distinctions shall be considered in the representation (e.g., data, models, diagrams) of the concepts in concern.

The architecture of MPMS, whose final result is presented in this thesis, is designed in an iterative fashion, following the design cycle of Wieringa (2009). More specifically, the initial version has been designed to fit the HORSE framework, which is a reference architecture for cyber-physical systems in smart manufacturing (Traganos et al., 2021). Since the HORSE architecture has undergone various iterations (documented in the project's deliverables and summarized in Grefen & Boultadakis (2021)), the MPMS architecture has been reviewed as well to stay in synch with the former. The logical architecture has also been applied in the OEDIPUS project with different realization systems (as presented in Section 7.1.2), enhancing in that way its generalizability. Finally, the MPMS architecture has been refined to fit the SHOP4CF architecture (Zimniewicz, 2020), which is an adaptation and extension of the HORSE architecture. Apart from the design iterations with respect to CPS reference architectures, extra design iterations have also been applied for the logical software architecture, as further elaborated in Section 5.3.3.4.

5.3.2 WFMS/BPMS reference architectures

As this research advocates the application of BPM theories and technologies in smart manufacturing, a BPMS should be the basis for designing a system for operations process management. Thus, the design of MPMS shall not start from scratch, but instead, it should be based on existing (reference) architectures. Three of these reference architectures (RAs) have been studied and are briefly introduced in the following three subsections.

5.3.2.1 Workflow Reference Model (WfRM)

The Workflow Reference Model (WfRM) (Hollingsworth, 1995), defined by the Workflow Management Coalition (WfMC) (Palmer, 2009), is a long-established reference architecture for WFMS. The model is often used as a basis for designing a BPMS (Pourmirza et al., 2017). It is a high-level system aspect model, describing the main components of a WFMS and the interfaces among those. The WfRM is illustrated in Figure 84.

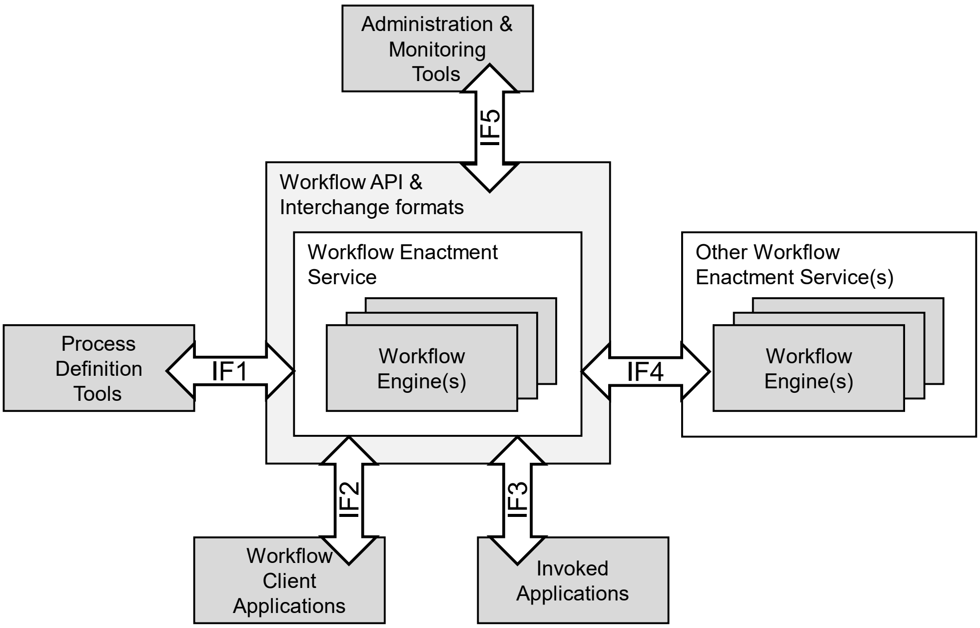


Figure 84: Workflow Reference Model (WfRM) (Hollingsworth, 1995).

The heart of WfRM is the workflow enactment service, where one or more workflow engines execute specified workflows. The workflow enactment service is wrapped in an interface layer, which connects the workflow engine(s) to other modules through interfaces (IF) 1 to

5. The process definitions tools are used to specify workflows (business processes). The workflow client applications are software modules to allocate tasks to human actors. Comparably, other invoked applications are used to automatically perform tasks in a workflow. A workflow engine can interact with other workflow engines in distributed workflow management environments. Finally, administration and monitoring tools provide support for the operational management of the enactment service.

5.3.2.2 Mercurius reference architecture

Another reference architecture for WFMS is that developed in the Mercurius research project (Grefen & Remmerts De Vries, 1998). The high-level model is shown in Figure 85. This is also a system aspect model, but with explicit link to data aspect architecture (by showing how data stores are linked to system modules).

The high-level model consists of three main modules (highlighted in grey) providing support for workflow design (Workflow Design), workflow enactment (Workflow Server) and end user functionality (Workflow Clients). The design data are stored in a database and are used by the workflow server. The workflow server interacts through communications systems(s) (CS) with other workflow servers. It also interacts with applications systems (AS) or operating systems (OS). The user interface systems (UIS) provide interactive functionality for end users. The three main modules often run in different environments and platforms, the boundaries of which are indicated by the dotted lines.

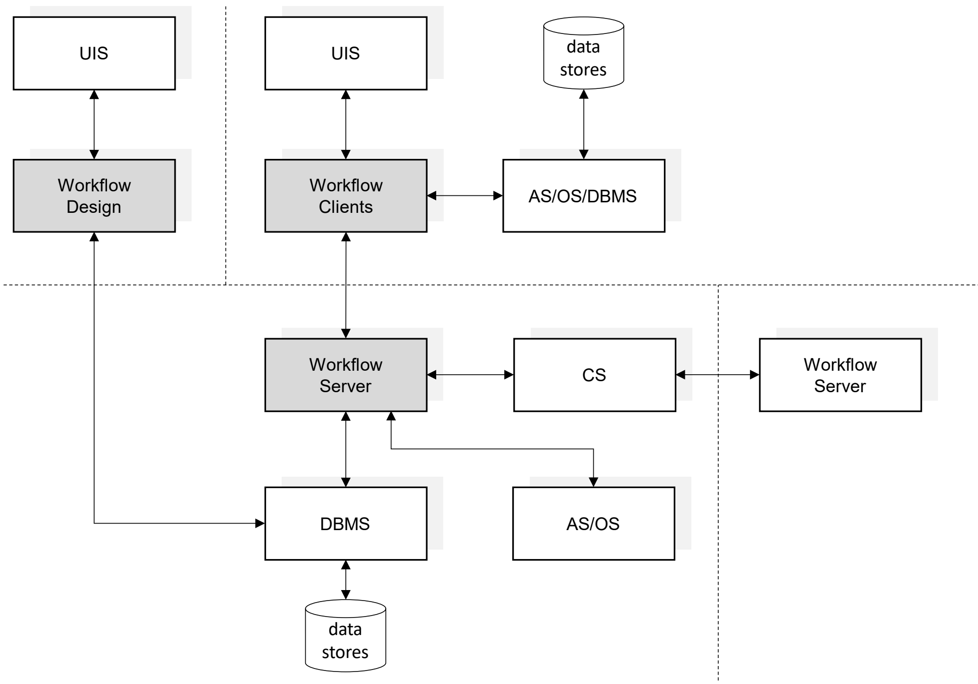


Figure 85: Mercurius Reference Architecture for WFMS, high level (Grefen & Remmerts De Vries, 1998).

The high-level model of Figure 85 is further elaborated in various aggregations levels per main module. For sake of brevity, the refinements are not shown here but they are taken into account in the design of MPMS per need (i.e., are referred and discussed when appropriate).

5.3.2.3 BPMS Reference Architecture (BPMS-RA)

A recent reference architecture for BPMS, called BPMS-RA, has been proposed by (Pourmirza et al., 2019) to cover latest developments on software and system engineering, such as the service-oriented paradigm and the data analytics. BPMS-RA has been designed based on analysis of literature on reference architectures (such as the WfRM and the Mercurius architecture) and of existing commercial implementations. In a high-abstraction level, it consists of three main components:

i. SOA-WfMS (Service-Oriented Architecture – Workflow Management System) component, offering functions such as business process modeling and execution;
ii. BPI&BPA (Business Process Intelligence & Business Process Analytics) component, monitoring and controlling BPMS-RA-compliant systems
iii. AAA (Authorization-Authentication-Accounting) component, providing security functions.

Figure 86 presenets an overview of the BPMS-RA, decomposing each of the main components into classes of subcomponents to procide specific functionality. Adjustanecy of components (shared borders) represent interfaces amone them (whose illustration is ommitted for sake of brevity).

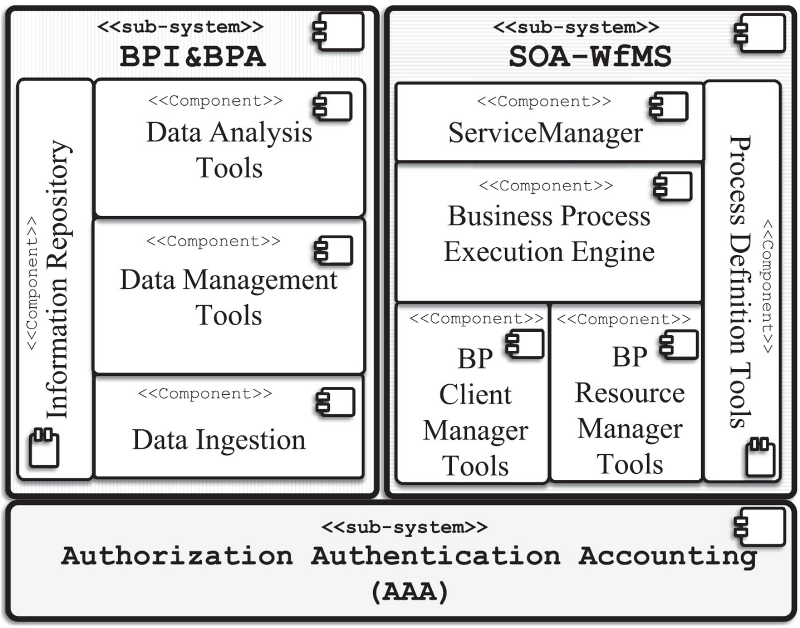


Figure 86: Novel BPMS Reference Architecture (BPMS-RA) (at aggregation level 2) (Pourmirza et al., 2019).

5.3.3 Logical view of MPMS

The logical architecture of MPMS is viewed from the five aspects of UT5 (Figure 82). Four of the aspects are discussed individually in the following four subsections. The fifth one, the platform aspect, is discussed in Section 5.4, after MPMS has been positioned in the context of a CPS.

5.3.3.1 Organization aspect

MPMS, as a centralized process orchestration system, shall provide global control of processes, either business or manufacturing ones. A manufacturing enterprise, which is typically organized according to the physical hierarchy of the IEC62264-1 standard (Figure 18), requires different types of control at different levels (in respect to the global-local dichotomy design principle). At enterprise, site, area and work center levels, global control is needed. On the other hand, at work unit level, local control is adopted. An illustrative physical hierarchy of an enterprise, indicating the difference on the types of control per level, is shown in Figure 87.

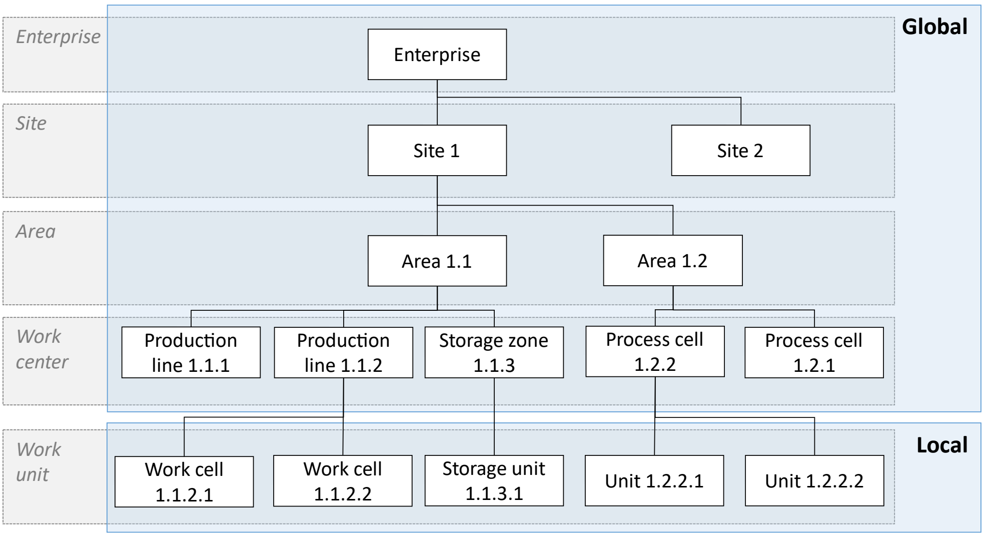


Figure 87: Example of a physical hierarchy of a manufacturing enterprise (based on IEC62264-1 standard), depicting the distinction on the global-local control regimes.

Consequently, MPMS shall provide process support across all work centers (i.e., production lines, process cells and storage zones) in a factory. The support has various forms:

- Selecting the right work center to produce a product (possibly through an MES). The "right" can be based on resource availability, product and equipment specification, maintenance schedules, etc. That means information from MES, CMMS or other relevant systems might be needed.
- Assigning the right resources to the right locations to perform activities. This also involves transferring resources from on work center or work unit to another (e.g., an operator or a mobile robot that can be shared between two work cells in the same or even different production lines).

- Having a clear overview of the production status across all work centers (and their work units). The status can refer to either normal/expected outcomes, or exceptional events/situations.

MPMS, through global control, is enabler for horizontal process integration.

5.3.3.2 Process aspect

Advocating that a centralized process management system can orchestrate both business and manufacturing processes demands for a clear process architecture. Given that a holistic view is required across various functional and physical levels, the process architecture gets fairly complex. A top-down design approach is followed to deal with this complexity. The design starts at the level of end-to-end, enterprise business processes. With stepwise refinements, it moves to the levels of manufacturing processes, manufacturing tasks and steps.

The end-to-end process model includes activities for designing a product, designing the production process, selling the product, purchasing the material to produce it, manufacturing it according to the design specifications, and delivering it to customers. Of course, the sequence of activities varies, depending on the engineering and production strategies. In standardized production, products are manufactured based on fixed product designs. Thus, a typical lifecycle of activities starts with one-time product and process design, setting up the production, and then, in a recurring way, selling the product, buying required material (making the assumption of zero-stock manufacturing for sake of simplicity), actual production and delivery. Such an example is illustrated in Figure 88. The process design and manufacturing activities are highlighted in blue, as the main focus of process management.

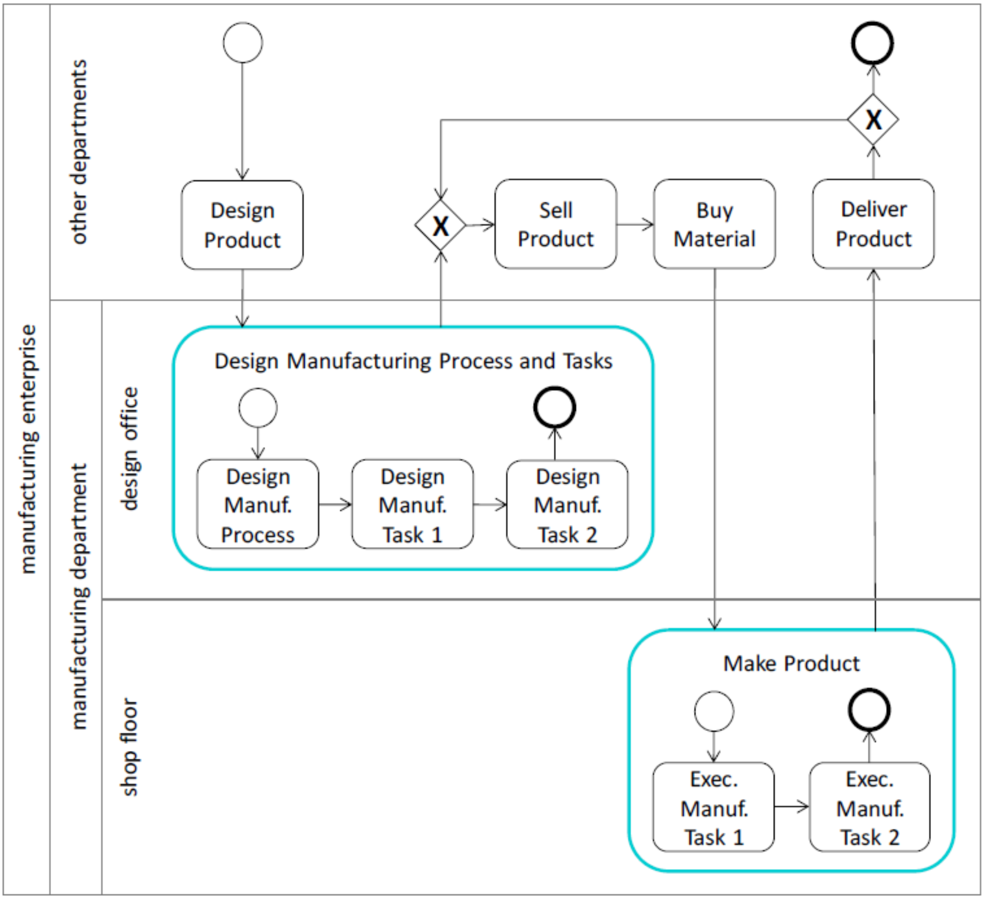


Figure 88: Illustrative high-level enterprise process for standardized production (the blue-highlighted subprocess are the main focus of MPMS) (Grefen & Boultadakis, 2021).

In customized production, a product is designed or adapted based on customer's specifications. Of course, selling the product or establishing a contract precedes the design. The actual manufacturing follows the design. An illustrative example is shown in Figure 89 (for reasons of brevity, the design and make product subprocesses are placed within the single manufacturing department and not split into design office and shop floor, like in Figure 88).

Other type of production strategies or a mix of them can be adopted by manufacturing enterprises. High-level process models can be drawn accordingly. What is most important to note here, is the separation of activities between design and execution phases (according to the design-execution dichotomy design principle) and the refinement of processes into tasks. This refinement has already been included in the examples of Figure 88 and Figure 89, and further refined below.

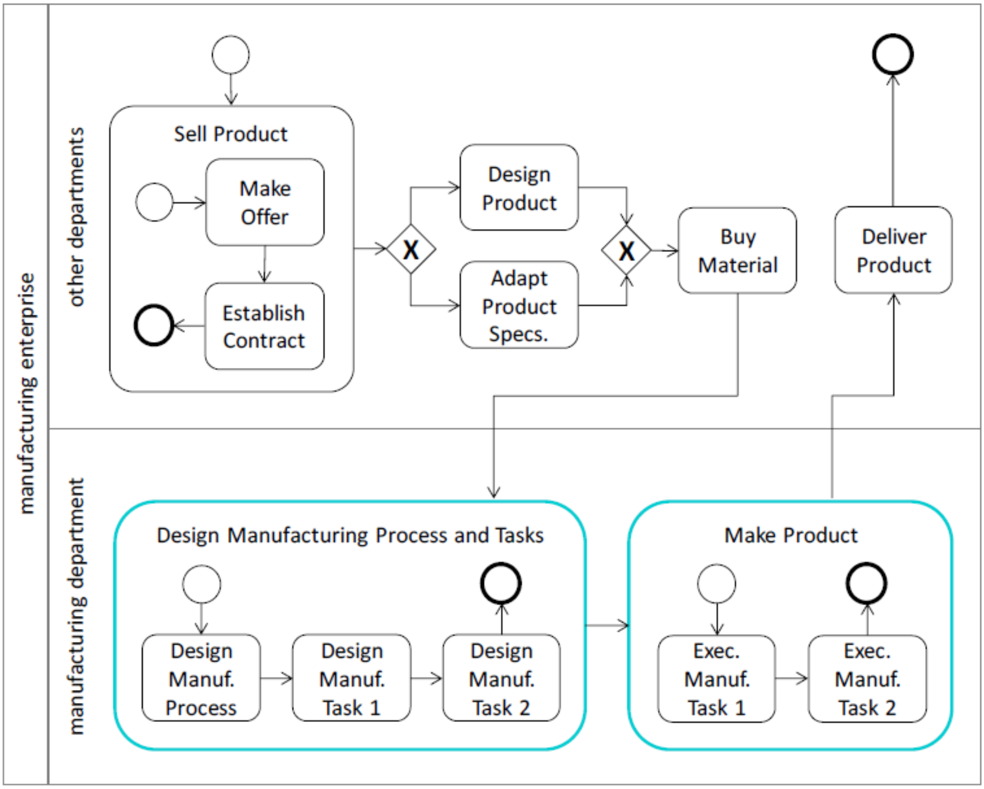


Figure 89: Illustrative high-level enterprise process for customized production (the blue-highlighted subprocess are the main focus of MPMS) (Grefen & Boultadakis, 2021).

The high-level enterprise process models of both Figure 88 and Figure 89 contain activities within the manufacturing department, which constitute the manufacturing process model. In standardized production, the design activities and those related to the execution of the modeled process for actual manufacturing are decoupled. They take place independently, in different phases and in different frequencies. The design is typically performed once, while the product based on it is instantiated multiple times. An example is illustrated in Figure 90 (with the "loop" marker in execution activities denoting the multiplicity). In customized production, design and execution activities are organized in a sequential fashion, as shown in Figure 91.

Figure 90 and Figure 91 show the further refinement of manufacturing tasks design and execution steps respectively. Designing a process requires determining the right sequence of tasks, performed by eligible resources at the right place, including any business rules. Designing a manufacturing task requires determining its input and output parameters, the location it takes places, the needed roles of resources to perform it with the right skills, abilities, authorizations, etc. Tasks are further refined into steps for more detailed elaboration. In both Figure 90 and Figure 91, the design of manufacturing tasks and steps is performed in a sequential way, but parallel activities are possible as well.

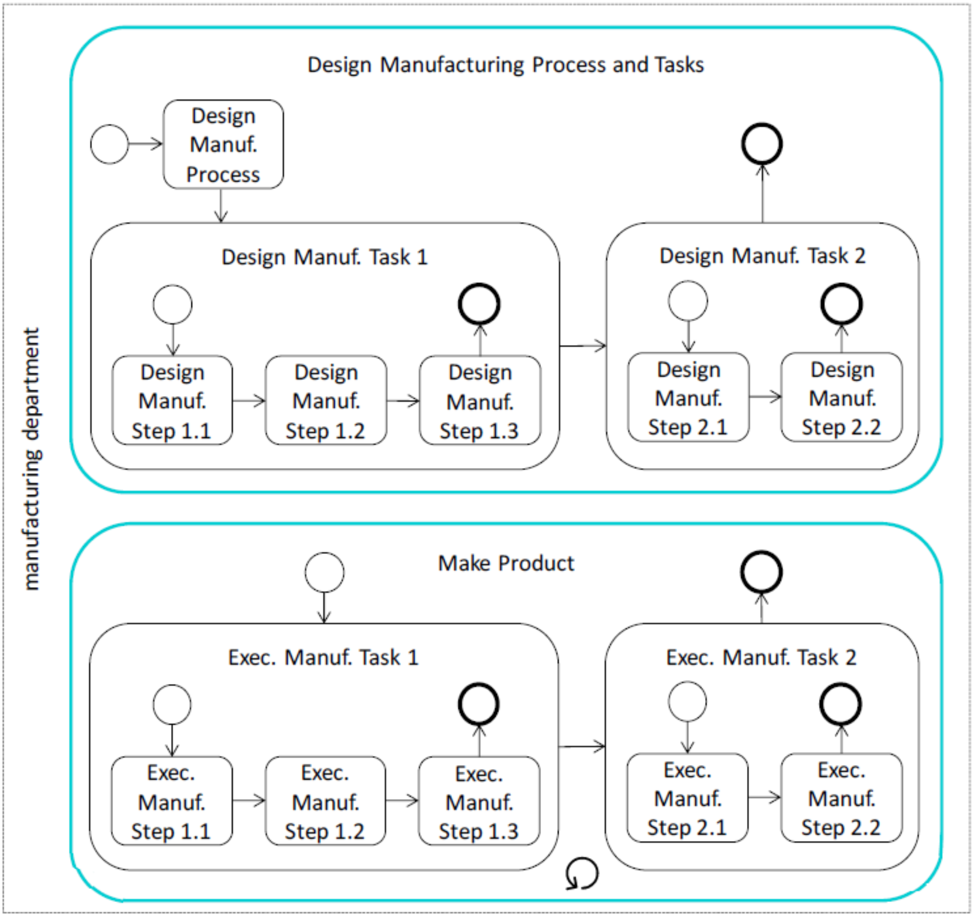


Figure 90: Illustrative manufacturing process for standardized production, with refined tasks and steps (Grefen & Boultadakis, 2021).

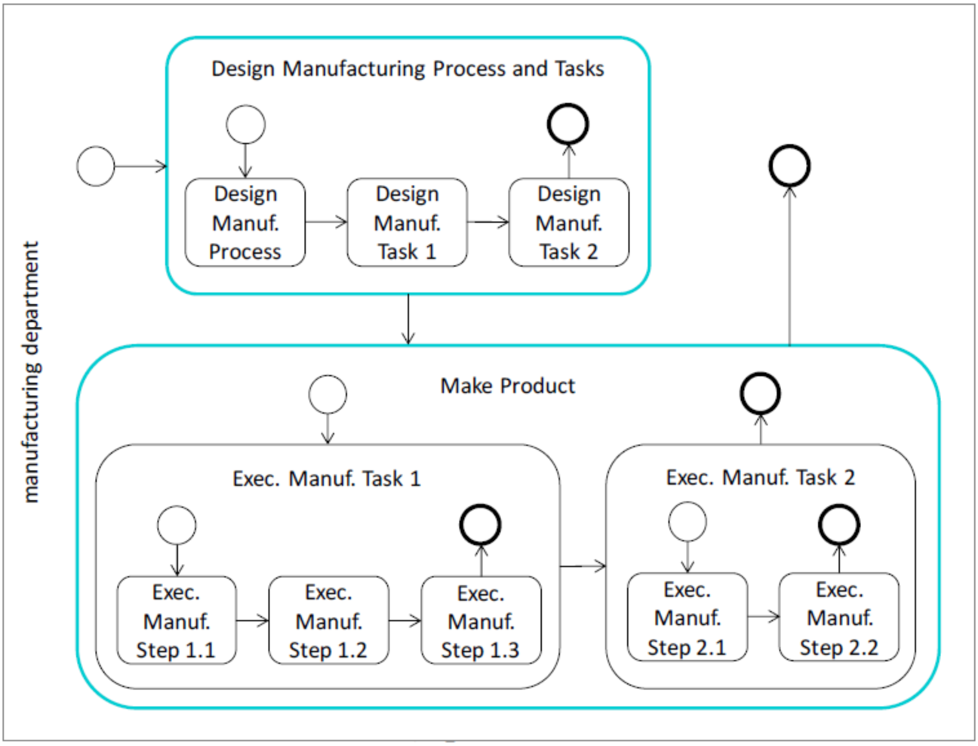


Figure 91: Illustrative manufacturing process for customized production, with refined task and steps (Grefen & Boultadakis, 2021).

Steps can also be refined in sub-steps, for the lowest level of elaboration, as shown in Figure 92. Note that the design and execution of steps and sub-steps is not responsibility of MPMS, however it is discussed for sake of completeness. Other systems play the role of orchestrating activities on step level, following the global-local separation principle.

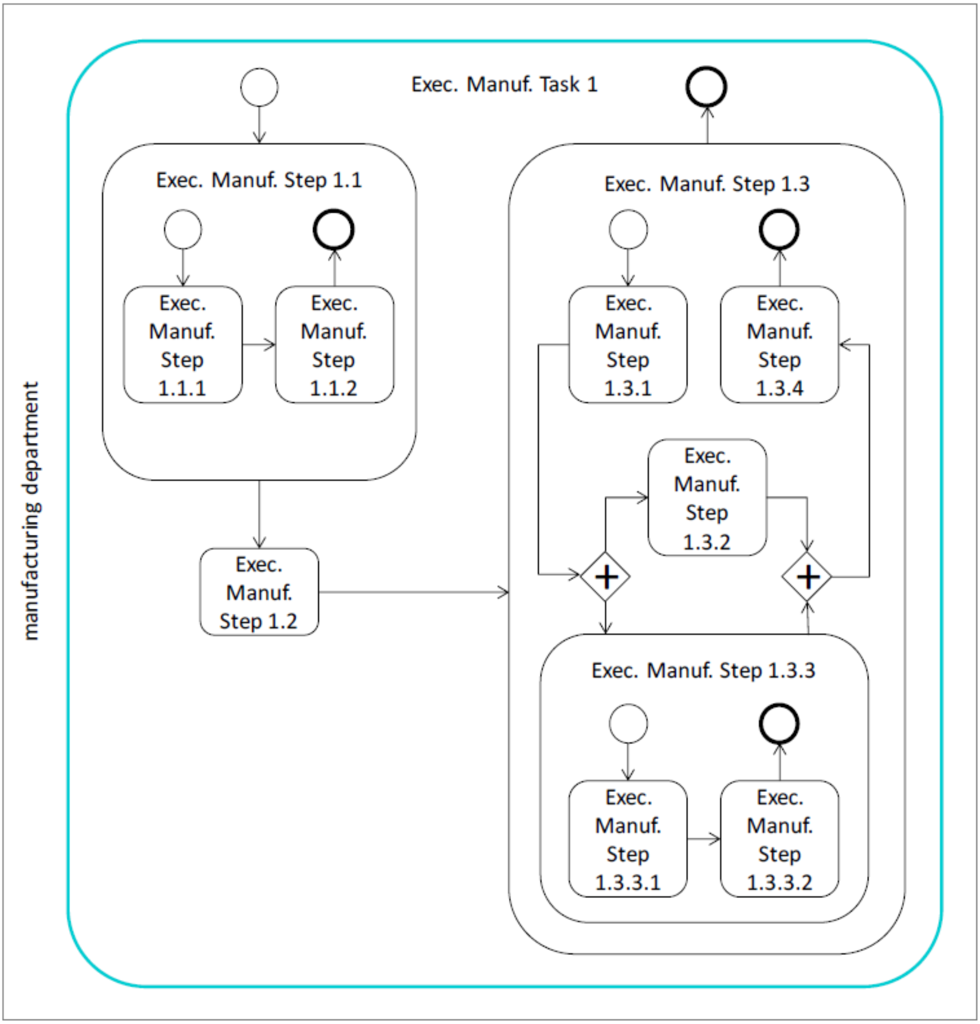


Figure 92: Illustrative manufacturing task with refined steps and substeps (Grefen & Boultadakis, 2021).

An example of a task refinement into steps and sub-steps (taken from the CPP pilot case) is shown in Figure 93. The activity of unloading paper from the high-capacity stacker (HCS) of a printer onto a work-in-progress (WIP) tray, performed by a mobile robot with a robotic arm mounted on it, is modeled as a single task. The only input parameter of the task is the signal from the HCS that paper is ready for unloading, and the only output parameter is the confirmation that the paper has been deposited on the tray. On a step level, the task is refined into a sequence of actions. The mobile robot has to first go in front of the HCS, then the robotic arm with a gripper mounted has to grasp the paper and deposit it onto the tray. The grasping part, is further detailed into lowering the gripper, closing it, and raising it back onto a position suitable for the next step. Note that the (sub-)steps are illustrated differently than the task activity to highlight that their modeling is typical done in a different notation (than the ones used for tasks).

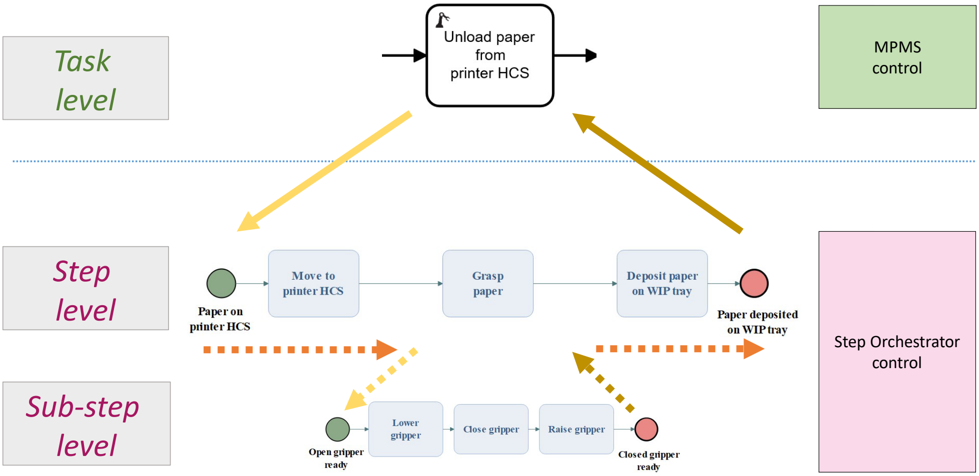


Figure 93: Example illustrating the refinement of a task into steps.

The process aspect architecture highlights the following points for process support by MPMS:

- Separation of business and manufacturing activities within a manufacturing enterprise.
- Separation between design and execution activities.
- Hierarchical and layered design/modeling.
- Distinction on task-step level for clear separation between global/local control.
- Support on processes/activities with different lifecycles/time horizons and different cardinality.

5.3.3.3 Data aspect

The data aspect architecture is extremely relevant, as it is used to represent both the physical and the functional entities in a way that these are usable by MPMS and other (information) systems. The well-established formal approach by West, (2011), summarized in Figure 94, is followed. The design starts with data requirements, which, here, take the form of context information on smart manufacturing. Based on these inputs, concept data models are designed in order to define entities and their relationships. The technical representation of a concept data model depends on specific technical constraints (e.g., a selected middleware technology can dictate the technical format of data). As this chapter views MPMS from a logical perspective, concept data models are designed here, leaving discussion on technical details for Section 6.2.3 (in which the development view is discussed).

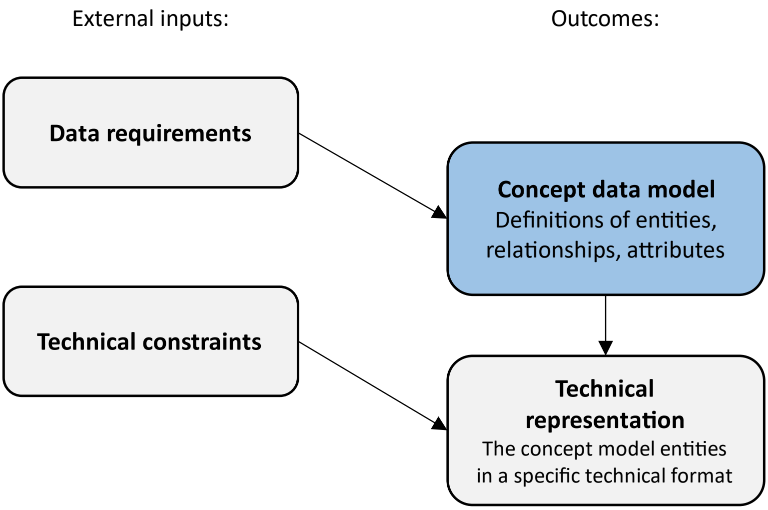


Figure 94: Data modeling approach, with the focus on concept data models in this section (from (West, 2011)).

As MPMS covers many aspects of smart manufacturing, a set of partial concept models that each covers an important aspect is designed at first. The individual concept models are then integrated into an overall concept model that relates all important aspects to each other. The individual concept models are:

- an *activity* concept model, which specifies the concepts and their relations related to activities performed in smart manufacturing context;
- an *agent* concept model, which specifies the concepts and their relations related to agents (actors) that perform activities in smart manufacturing context;
- a *resource* concept model, which specifies the concepts and their relations related to any type of resource appearing in smart manufacturing environments;
- an *event* concept model, which specifies the concepts and their relations related to events occurring in smart manufacturing context that require reactions;
- a *location* concept model, which specifies the concepts and their relations related to the physical environment in which activities are executed.

The concept data models are discussed below and represented in UML class diagrams. Note that the presented models are the consolidation of the ones applied in HORSE, OEDIPUS and SHOP4CF project following the iterative design process as mentioned in Section 5.3.1.

5.3.3.3.1 Activity concept model

The process aspect architecture makes clear what the hierarchy of the activities in a manufacturing enterprise is. The high-level activity data model is illustrated in Figure 95 and explained below.

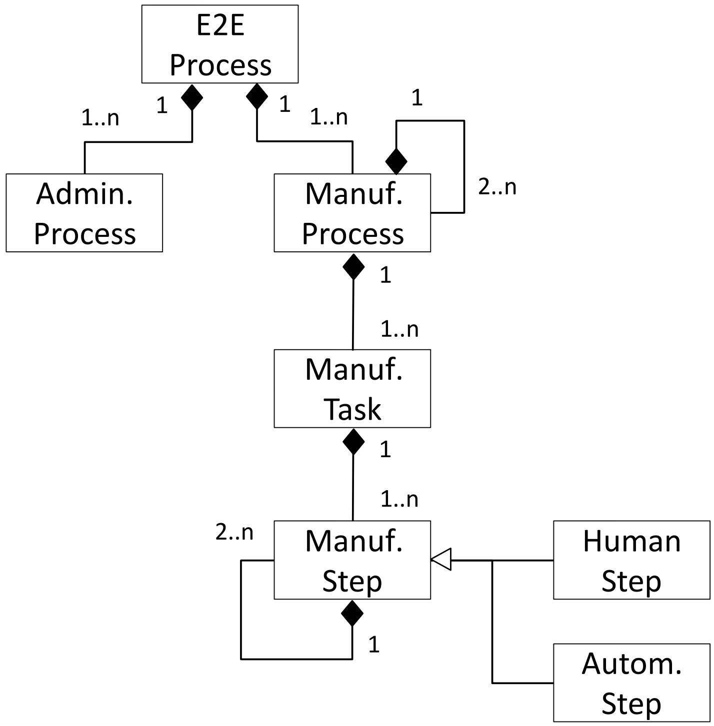


Figure 95: Activity concept model.

At enterprise level, an end-to-end (E2E) process (i.e., from order reception until delivery or after-sales services) consists of both administrative (e.g., sales, invoicing) and manufacturing (i.e., production, warehousing) processes. Considering that an E2E process is instantiated for each individual customer order, there is at least one administrative process and at least one manufacturing process. More than one manufacturing processes exist in an E2E process in case a customer order is not fulfilled in one production run or the order consists of different product types.

A manufacturing process can be layered, i.e., it can exist of two or more sub-processes (indicated with the self-relation). A bottom-level manufacturing process consists of a number of tasks. The concept of manufacturing task refers to the action(s) performed to complete a production goal (according to Luck (1995), an action is defined as a discrete event that changes the state of the environment). Tasks cannot be layered, i.e., only a single layer exists. This leads to a clear control and actor allocation mechanisms, with MPMS having a clear responsibility on task level (as also illustrated in the example of Figure 93). A manufacturing task consists of a number of steps, as the single actions to be performed by a (team of) agent(s). As the agents (actors) can be either humans or automated devices (as discussed later in the agent concept model), steps are further specialized in human steps and automated steps. This is necessary since these two types of steps are designed and performed differently (a human operator is instructed differently than a robot). Steps may consist of sub-steps, i.e., they may be hierarchically organized in more detailed activities (as also illustrated in the example of Figure 93).

Considering the design-execution dichotomy design principle, each of the above entities should account for both design and execution phases. Focusing on the three main entities – processes, tasks, steps –, the activity concept model results in the one shown in Figure 96. During actual execution of activities, an instance of a process, task, or step is created according to its corresponding definition.

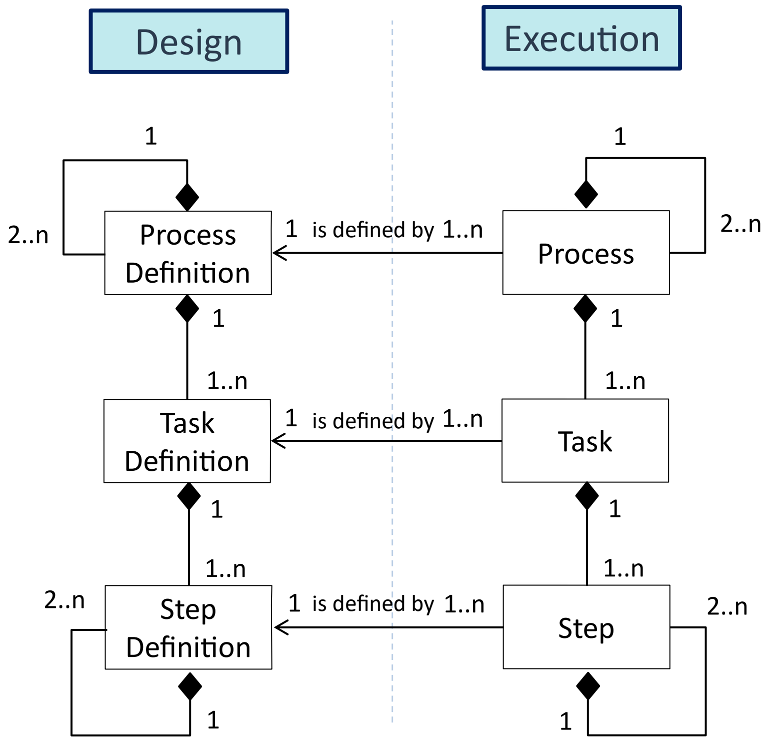


Figure 96: Activity concept model with respect to design-execution design principle.

5.3.3.3.2 Agent concept model

Activities are performed by agents. An agent is defined as *an instantiation of an object together with an associate goal or set of goals* (Luck, 1995). Simply, an agent is an actor that performs a step (work) in a manufacturing process. The agent concept model is presented in Figure 97 and explained below.

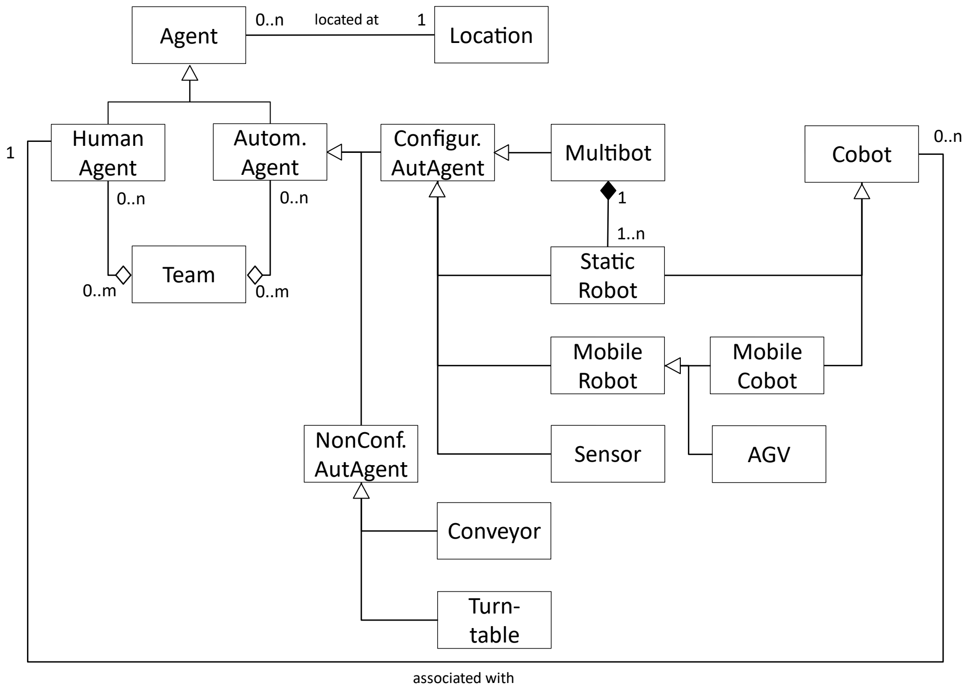


Figure 97: Agent concept model.

The agent is further specialized into human or automated agents, with each of these two types having specific characteristics that the general type does not have. Agents form team(s) and the concept of teams might have specific composition requirements. For instance, the team shall consist of at least one human operator. Thus, the team concept is linked to the agent subclasses and not to the superclass.

An automated agent is any kind of equipment/device that can autonomously perform an action. The concept is further specialized into configurable automated agents (e.g., a programmable robot) and non-configurable automated agents (non-programmable active manufacturing equipment, e.g., a conveyor belt). A configurable automated agent is further specialized into static robots (i.e., placed at a fixed position in a work cell), mobile robots (i.e., capable of moving around at the shop floor), with AGV a specialized form of mobile robot, and sensors, as devices that can sense the environment. Combining static robots into one entity, form the concept of multibot. A cobot is a kind of configurable robot, either static or mobile, that collaborates or works together with a human operator and thus, is linked to the human agent. A human can be associated with more than one robot.

Any agent, either static or moveable, is linked to the concept of location, as its position is relevant for various reasons. For instance, the location is a criterion during the allocation of an agent on a task, especially given the dynamicity of changes on a shop floor.

5.3.3.3.3 Resource concept model

Agents, humans and robots, are the "active" resources of an enterprise. Other types of resources, though, are relevant in smart manufacturing context and need to be represented.

For example, a gripper, as a physical asset, combined with a robotic arm, compose a set of resources that are eligible for performing various tasks. Accordingly, the information regarding a tool, is relevant when instructing an operator on how to perform a manufacturing step. The resource concept is an abstract entity, i.e., it does not exist directly, generalized as one of its concrete subtypes. These are: agent, material, and (physical) asset, as shown in Figure 98.

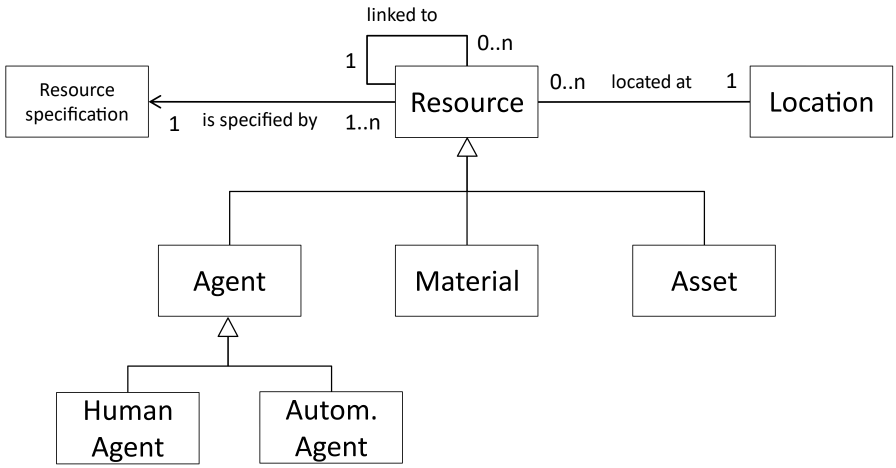


Figure 98: Resource concept model.

The agent type of resource has already been discussed. The material concept refers to a product (final or intermediate) or an ingredient used in a manufacturing process. Its attributes may consist of physical states, e.g., dimensions, or process-related information, e.g., the result of an inspection. The asset concept refers to tangible objects that are neither agents nor material. Typical types of assets are the tools or elements of some equipment. All types of resources are linked to the location entity.

A resource can reference other resources that are physical linked to. For instance, a robotic arm mounted on a mobile robot can be modeled as two separate resources but are linked to each other. This information is useful during allocation mechanisms, e.g., when the robotic arm is operating an action, the mobile robot on which it is mounted should also be considered as busy and not available for a new operation.

The resource entity is specified by a resource specification entity. The latter is needed in the design phase in which resources are specified as abstract classes. The classes are then instantiated with detailed information during execution. For instance, a resource “Operator John” is an instantiation of the resource specification “Operator”.

5.3.3.3.4 Event concept model

The event concept model is shown in Figure 99. An event is linked to its use, in which it is processed, e.g., to make a decision or to store data in a log. The event is specialized into alerts, measurements and notifications. An alert is an event generated in an exceptional situation. A measurement is a planned, periodic event. A notification is a general event about something happening. A notification about an order or an external event fall under that category. Alerts should not be confused with the exceptions, as discussed in Chapter 4. Even a measurement, when exceeds a threshold, is considered as an exception. The discussion here is about the event concept, as a type of important information to be represented in a smart manufacturing context. Section 6.2 elaborates on the data model on exceptions.

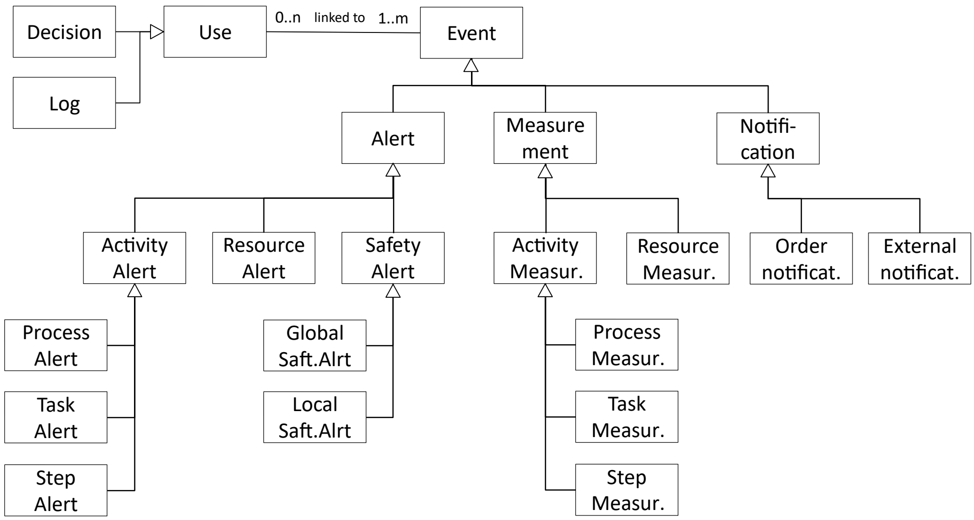


Figure 99: Event concept model.

Alerts are specialized as activity, resource and safety alerts. An activity alert is generated when an activity is being executed. This happens at process, task, or step level, following the hierarchy of the activity concept model. An example of a step alert is a timeout on its predefined execution time.

A resource alert is generated by any type of resource of the resource concept model. These alerts should be independent to the activity the resource is performing at that moment. For example, a low battery level alert generated by an AGV is a resource-related alert, irrespective of whether the AGV is moving or not.

A safety alert is generated by an observed safety breach, either at a global level (i.e., site, area, or work center) or local level (i.e., work unit). A typical example of a safety alert can be a fire at a production area.

Regarding the measurement concept, it is further specialized into activity and resource measurements. A performance measurement is typical example of an activity measurement. A resource measurement refers to state of a resource at a given time. Typical values can be available, busy, under maintenance, etc.

5.3.3.3.5 Location concept model

The location concept is used to represent the data related to the organization aspect architecture. It represents a specific place in a factory, or it can also represent structures of the enterprise as entities, following the physical hierarchy of the IEC62264-1 standard. The location concept model is shown in Figure 100.

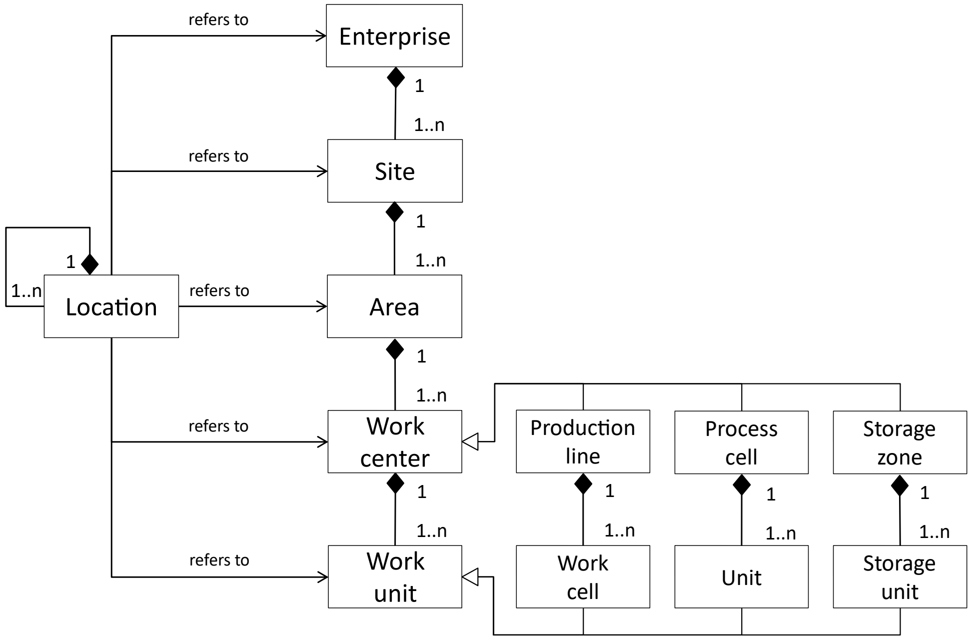


Figure 100: Location concept model.

The location has an important function: it can either be source or target. This is useful when tasks are linked to a location. For example, a transportation task for an AGV to move to a storage zone, requires setting the location of the storage zone as a target. Accordingly, when the AGV leaves the storage zone for a next target, that storage zone is considered as source from the AGV's task point of view.

5.3.3.3.6 Integrated concept model

Combining the five partial concept models results in an integrated concept model. A high-level overview of this integrated concept model is shown in Figure 101. The partial concept models have been simplified to not make the overview diagram overly complex. It is the links between the partial concept models that is important to discuss.

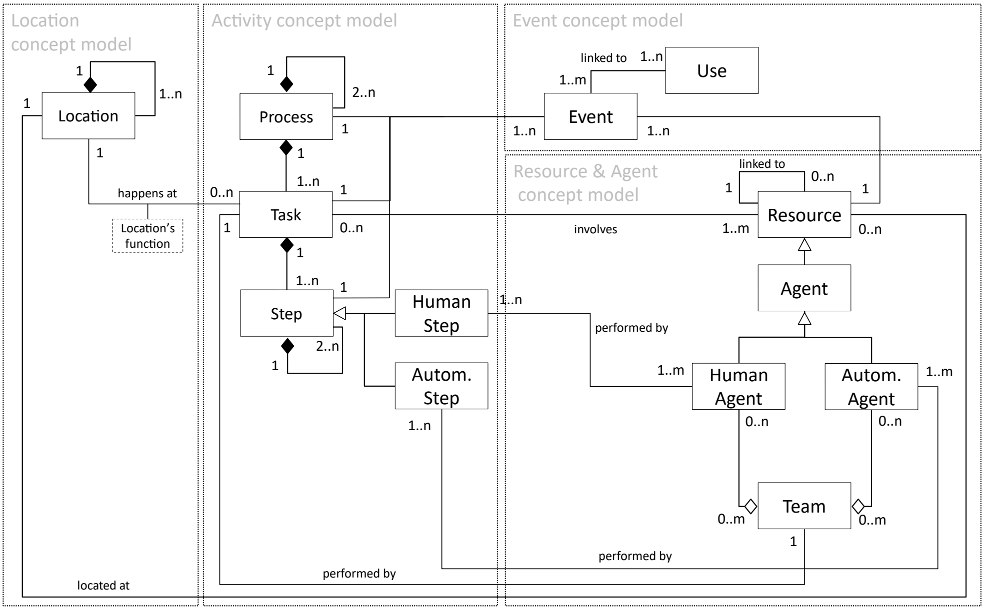


Figure 101: High-level overview of integrated concept model.

The link between the activity concept model and the agent concept model is established in two ways. Firstly, a task is performed by a team of agents. Secondly, a step, as an individual activity, is performed by a specific agent. Linking the activity concept model and the resource concept model is established at the task level with the "involves" relationship. Each task involves a number of resources to be performed. For reasons of brevity, only the agent type of resources is shown, but apparently the task might involve material or physical assets. It should also be noted that the linking between the task and resource concepts is kept simple here. A more enhanced linking includes the concepts of role and additional attributes, as shown in Figure 102, relevant for advanced agent allocation mechanism (Erasmus, 2019).

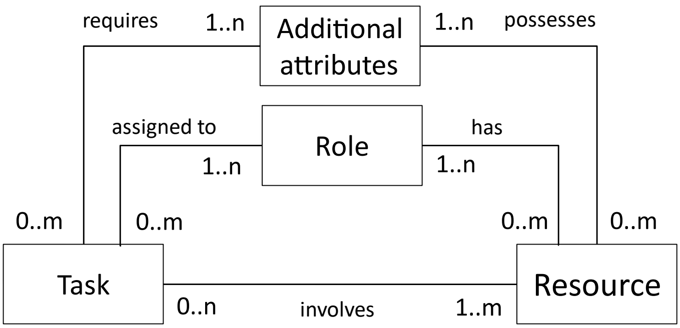


Figure 102: Enhanced relationships between Task and Resource concepts (left out of scope in the integrated concept model for simplicity reasons) (Erasmus, 2019).

The link between the activity concept model and the location concept model is established at the task level with the "happens at" relationship. The location's function (i.e., source, target) is important to orientate activities (as has been discussed in partial location concept model).

The link between the activity concept model and the event concept model is relatively simple. Events, be it alerts or measurements, are linked to any level of activities. Events are also linked to resources, referring to either alerts or measurements.

Finally, the link between the resource concept model and the location concept model is established with the "located at" relationship.

5.3.3.3.7 Mapping to the IEC62264-1 standard

The concept models presented above have been designed based on information needed to represent the physical and functional entities in a smart manufacturing context. To safeguard the rigor of the designed models and enhance their interoperability with existing approaches and standards, the concepts are confronted with the ones defined by the IEC62264-1 standard. The standard describes concepts in a manufacturing context, though without focus on smart manufacturing. Table 19 provides an overview of the mapping between the concepts.

Table 19: Mapping of designed concept models to concepts from IEC62264-1 standard.

Designed concepts	**IEC62264-1 concepts**
Process definition Task definition Step definition	Process segment
Process	Operations definition
Task	Operations segment
Resource specification	Equipment/Personnel/Material/Physical Asset specification
Human agent	Personnel
Automated agent	Equipment
Material	Material
Asset	Physical asset
Alert	Work alert
Location	Equipment hierarchy

5.3.3.4 Software aspect

As already mentioned, the design of MPMS is based on existing reference architectures of BPMS. The WfRM and the Mercurius reference architecture have been used as starting points to design the logical software architecture of MPMS. In an iterative design fashion (Wieringa, 2009), the design has been adapted to take into account the BPMS-RA, as a contemporary novel reference architecture for BPMS that has emerged recently.

MPMS shall satisfy specific requirements for process management in smart manufacturing. Therefore, it should be checked whether the reference architectures, used as basis for designing the MPMS, satisfy those requirements. Table 20 checks the coverage of the requirements listed in Section 2.3, grouped as general system functions, by the three selected reference architectures, with the purpose to identify existing, missing or limited functionality that shall be enhanced for application in smart manufacturing.

Table 20: MPMS requirements coverage by BPMS reference architectures.

R#	Requirement	System function	Coverage by WfRM	Coverage by Mercurius	Coverage by BPMS-RA
Design					
R01	The MPMS shall provide modeling support of complex processes that involve synchronization of activities by various actors (including human-robot collaboration scenarios)	Process definition	Limited – Process Definition Tools cover (business) process modeling but no explicit support for physical manufacturing processes.	Limited – Workflow Design module > Workflow Design Engine > Global Design and Detail Design modules cover (business) process modeling but no explicit support for physical manufacturing processes.	Limited – Process Definition Tools > Business Process Modeling covers (business) process modeling but no explicit support for physical manufacturing processes.
R02	The MPMS shall be able to define manufacturing resources, such that it can determine during execution which resource should perform an activity (linked to R06).	Resource definition	Missing – No explicit module to define resources.	Limited – Workflow Design module > Workflow Design Engine > Organization Design can be used to define resources. Though, as the module specifies mostly human resources, extra coverage for automated resources is needed.	Limited – BP Resource Managers Tools are used to specify resources that can perform activities in business processes. Though, as these refer mostly to human resources, extra coverage for automated resources is needed.
R03	The MPMS shall be able to define tasks, such that clear control of activities is provided in both modeling and execution phases.	Task definition	Missing – No explicit module to define tasks.	Limited – No explicit module to define tasks. Workflow Design module >	Missing – No explicit module to define tasks.

				Workflow Design Engine > Detail Design could be used but without explicit definition on task level.	
R04	The MPMS shall be able to represent the physical equipment hierarchy, such that functional processes are mapped to their respective physical environment.	Location definition	Missing – No explicit module to define locations.	Limited – Workflow Design module > Workflow Design Engine > Organization Design can be used to define the organization structure, but extra support for defining (physical) locations is needed.	Missing – No explicit module to define locations.
Execution					
R05	The MPMS shall be able to enact the modeled processes in an automated way.	Process engine	Existing – Workflow Engine	Existing – Workflow Server > Workflow Server Engine	Existing – Business Process Execution Engine
R06	The MPMS shall be able to dynamically select and allocate the most suitable resource(s) to tasks, based on task requirements and resource capabilities.	Agent allocation	Limited – Resource allocation is typically performed through the Workflow Process Engine, but no explicit module exists for advanced allocation mechanisms during runtime.	Limited – Resource allocation is typically performed through the Workflow Server Engine, but no explicit module exists for advanced allocation mechanisms during runtime.	Limited – Resource allocation is typically performed through the Business Process Execution Engine, but no explicit module exists for advanced allocation mechanisms during runtime.

R07	The MPMS shall be able to send a list of tasks to be performed by each actor in the production environment, for a specific production order.	Task delivery	Limited – Through IF2, tasks can be delivered to Workflow Client Applications (so tasks can be executed by actors). However, only tasks to human actors or to services are covered, without explicit support for task assignment to automated agents or heterogeneous team of agents.	Limited – Workflow enactment client delivers tasks to (office) workers. No support for task assignment to automated agents or heterogeneous team of agents.	Limited – Through Business Process Client Tools, tasks can be delivered to actors. However, only tasks to human actors or to services are covered, without explicit support for task assignment to automated agents or heterogeneous team of agents.
R08	The MPMS shall be able to accept notifications from actors in the production environment regarding a change of manufacturing system status, including actors' availability and status.	Event handling	Limited – The Workflow Process Engine is able to receive events regarding (human) tasks (through IF2) and other information (through IF3), but there is no explicit support for manufacturing entities (e.g., notifications from physical devices).	Limited – The Workflow Server > Workflow Server Engine > Event receptor is able to receive events regarding (human) resources and services, but there is no explicit support for manufacturing entities (e.g., notifications from physical devices).	Limited – The Business Process Client Tools can receive events from resources, but no explicit support for events from automated resources. The BPI&BPA > Data Analysis Tools and the SOA-WfMS > Business Process Execution Engine are able to receive and process events, but there is no explicit support for

					manufacturing entities (e.g., notifications from physical devices).
R09	The MPMS shall be able to receive events regarding changes of the manufacturing system status.	Event handling	Limited – The Workflow Process Engine is able to receive events regarding (human) resources (through IF2) and other information (through IF3), but there is no explicit support for events from the manufacturing environment (e.g., events from physical devices).	Limited – The Workflow Server > Workflow Server Engine > Event receptor is able to receive events regarding (human) resources and services, but there is no explicit support for events from the manufacturing environment (e.g., events from physical devices).	Limited – The BPI&BPA > Data Analysis Tools and the SOA-WfMS > Business Process Execution Engine are able to receive and process events, but there is no explicit support for events from the manufacturing environment (e.g., events from physical devices).
R10	The MPMS shall be able to react on exceptional events that change the status of the manufacturing system	Exception handling	Missing – No explicit support for exception handling (exceptions are implicitly handled by the Workflow Enactment Service).	Limited – Exceptions can be designed in the Workflow Design module, but no explicit support for runtime handling. The Software bus manager > Protocol manager can handle only specific type of exceptions.	Limited – The Business Process Execution Engine > Exception Handling component handles exceptions. Support is needed for exceptions occurring in manufacturing environments.
R11	The MPMS shall be able to monitor the status of the manufacturing system during	Monitoring	Existing – Monitoring Tools. However,	Limited – Implicit monitoring functionality	Existing – BPI&BPA components.

	execution of processes.		monitoring of manufacturing resources shall be considered.	through Workflow Client > Extension Module. Monitoring of manufacturing resources shall be considered	However, monitoring of manufacturing resources shall be considered.
General					
R12	The MPMS shall be able to provide administration of processes.	Administration	Existing – Administration Tools	Existing – Implicit administration functionality through Workflow Server > Extension Module.	Existing – AAA component
R13	The MPMS shall be able to integrate to other EIS, including ERP/MES.	Interfacing	Existing – Interfacing through IF3 to invoked applications. However, no explicit support is provided for specific types of systems.	Existing – AS Interfaces.	Existing – Service Manager.

The coverage information, as briefly explained in Table 20, and the following design decisions yield the logical software architecture of MPMS, shown in Figure 103. New modules or modules that require extensions for application in smart manufacturing are highlighted in green, compared to the gray ones which require no adaptations (from a functionality point of view).

Design decision 1): The functionality of MPMS is grouped in two main classes: i) Definition Tools, covering R01-R04, ii) Process Enactment Service, covering R05-R10. The distinction respects design-execution design principle.

Design decision 2): Components for providing analysis functionality (as per BPMS-RA) are not included as no identified relevant requirement exists. Note that R11 for process monitoring mainly refers to static status provisioning during runtime, without explicit functionality for analysis. In that respect, a component Monitoring Tools (labeled per WfRM) is considered as an auxiliary component and not as an analysis component.

Design decision 3): Definition Tools include: i) Process definition, covering R01 (per all three RAs), ii) Resource definition, covering R02 (per Mercurius and BPMS-RA), iii) Task definition, covering R03 (new module), iv) Location Definition, covering R04 (partially per Mercurius).

Design decision 4): A Process Engine module covers R05 (per all three RAs).

Design decision 5): An Agent allocation module is added for explicit support for advanced allocation mechanisms by selecting the right agents to perform activities, covering R06 (in comparison to all three RAs which do not provide explicit support).

Design decision 6): A Task delivery module is added for explicit delivery of the right task information to the selected agents, covering R07. The module delivers tasks to either humans or auto agents (in comparison to all three RAs which deliver tasks to human operators or to call services to perform activities).

Design decision 7): The Agent allocation and the Task delivery modules are considered as part of Agent Tools. The term "agent" is selected over the term "client" of all three RAs for keeping consistency per agent and resource concept models.

Design decision 8): Interfaces between main classes and to external systems are highlighted keep the labeling of the WfRM.

Design decision 9): The Auto agent control system (outside the boundaries of MPMS) is not considered as an Invoked Application (per WfRM), but as an agent application (client application per all three RAs), and thus, is interfacing through IF2 and not IF3. This design decision is based on the fact that MPMS views automated agents as the counterpart of human agents, both of which perform activities in a manufacturing process.

Design decision 10): A Service/Integration layer is added to cover functionality of interfacing to agents (IF2) and other systems (IF3) (discussed in Section 5.4).

Design decision 11): Notifications from agents are received by the Agent Tools through IF2, covering R08.

Design decision 12): Events regarding agents are received by the Agent Tools through IF2. Events from other systems are received by the Process Engine through IF3. Both ways cover R09.

Design decision 13): An Exception handling module is added to provide functionality for reacting to exceptional events (R10) (per Mercurius and BPMS-RA). It is considered as an extension module of the workflow enactment server architecture of Mercurius (see Fig.7 of (Grefen & Remmerts De Vries, 1998))

Design decision 14): Monitoring tools (R11) are enhanced to satisfy the requirements of manufacturing environments compared to business environments.

Design decision 15): Administration Tools (per all three RAs) is added, covering R12.

Design decision 16): EIS and other (information systems) (outside the boundaries of MPMS) interface through IF3, covering R13 (seen as Invoked applications per WfRM).

Design decision 17): Data stores are omitted to keep the focus of the architecture on the functionality modules (per WfRM and BPMS-RA).

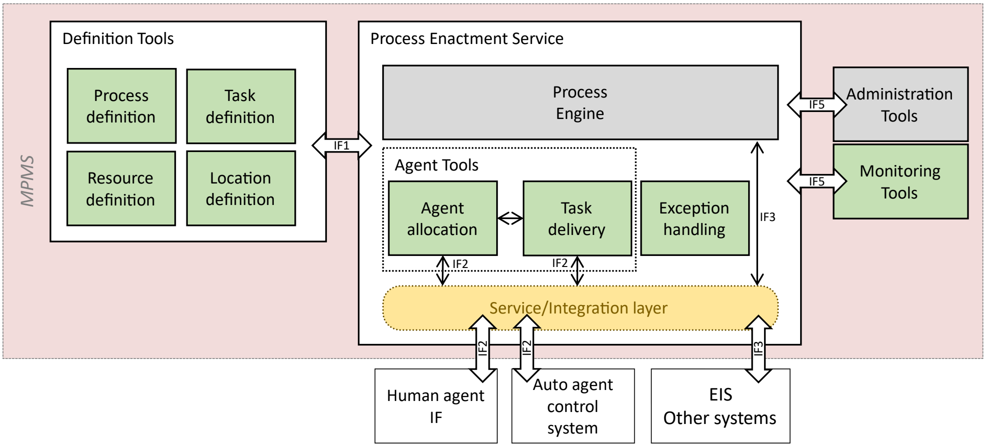


Figure 103: Logical software architecture of MPMS, with enhanced logical modules (highlighted green) and new logical modules (highlighted blue) for process management in smart manufacturing.

5.4 MPMS as part of a CPS

The architecture of MPMS has been discussed so far with a focus on the system's internal functionality. Main interfaces to other systems have been identified but have not been elaborated. To complete its specification, the system has to be placed in the context of a cyber-physical system architecture that provides horizontal and vertical integration in a smart manufacturing context. The logical software architecture of a CPS is first presented in Section 5.4.1. MPMS, as part of that architecture, interfaces to other systems. Integration to those systems through middleware technologies is elaborated in Section 5.4.2. Finally, Section 5.4.3 completes the specification of MPMS with the discussion of the platform aspect of the UT5 framework.

5.4.1 Logical software architecture of a CPS

The HORSE project resulted in a framework, as a reference architecture for cyber-physical systems that integrate smart technologies and provide manufacturing operations management in hybrid actors settings. The framework is a modular architecture with clear subsystems and interfaces at several levels of aggregation, resulting from a structured, hierarchical system design, based on theoretical principles and guidelines (Grefen & Boultadakis, 2021).

The HORSE system follows the two design principles of Figure 83, with clear separation between design and execution activities, for both global and local levels. The high-level logical software architecture is shown in Figure 104, labeled as aggregation Level 3 (Level 0 represents the HORSE system as a monolith, i.e., without internal structure, Level 1 the system after decomposition across Global/Local levels, and Level 2 after further decomposition across Design/Execution phases). The functional modules are briefly described below (Traganos et al., 2021):

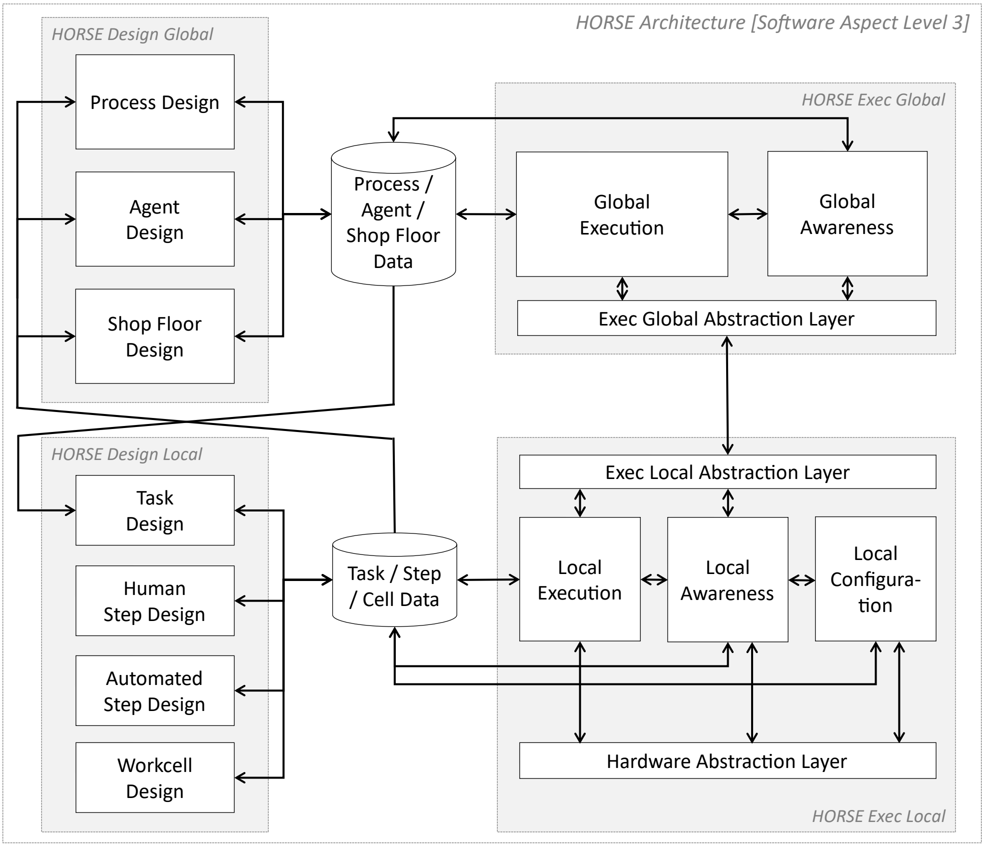


Figure 104: HORSE system high-level logical software architecture (at aggregation level 3) for cyber-physical systems in hybrid smart manufacturing (Grefen & Boultadakis, 2021).

<u>HORSE Design Global</u>

The HORSE Design Global subsystem covers functionality to design smart manufacturing processes at the global level, i.e., at the site, area and production line levels. There are three modules:

- *process design*, covering functionality to specify manufacturing processes in terms of process models, i.e., what is the sequence of activities and which agents are involved (specifications of roles);
- *agent design*, containing the functionality to specify the characteristics of agents. Agents can be humans, e.g., operators, or automated agents like robots;
- *shop floor design*, containing the functionality to specify the entire production area/site, both in terms of physical layout and safety aspects of all production lines/work cells and their inter-connections.

HORSE Exec Global

The HORSE Exec Global subsystem contains functionality to execute manufacturing activities across work units, i.e., at the site, area and production line levels. This includes two main modules:

- *global execution*, responsible to execute manufacturing processes. It retrieves process definitions from the process/agent/shop floor database that have been created by the design modules and automatically executes them in runtime;
- *global awareness*, monitoring the global state of the environment to guarantee the overall safety. It observes what is happening during the execution of the processes and in case of safety hazards communicates with the global execution module to interrupt them.

To make the implementation decisions regarding the HORSE Exec Global subsystem independent to the HORSE Exec Local subsystem, an abstraction layer (Exec Global Abstraction layer) that eases the communication of these two subsystems is included.

HORSE Design Local

The HORSE Design Local subsystem covers functionality to design manufacturing tasks at the local level, i.e., at the work cell level. The subsystem involves four main modules:

- *task design*, containing the functionality to design a manufacturing activity spanning a work cell, which can consist of multiple steps and which require agent(s) (human, automated, or a hybrid team of them) to execute them;
- *human step design*, covering the functionality to design and specify manufacturing steps that are performed by a human agent;
- *automated step design*, covering functionality to design and specify manufacturing steps that are performed by an automated agent;
- *work cell design*, containing functionality to support the physical design of a work cell

HORSE Exec Local

The HORSE Exec Local subsystem contains functionality to support the execution of manufacturing activities within individual work cells, i.e., at the work cell level. It includes three main modules:

- *local execution*, responsible to control the actual execution of manufacturing tasks and steps performed by (a team of) agents;
- *local awareness*, covering functionality to observe the physical status of a work cell, by receiving signals from sensors, cameras and human interface devices, analyzing them and notifying either the local execution module or the global awareness module;
- *local configuration*, containing functionality to configure resources on the physical shop floor, i.e., in the execution environment (as opposed to the functionality in the HORSE Design Local which typically happens in a design office). The configuration typically involves setting parameters that are very closely linked to the physical execution environment (such as teaching a robot by demonstration).

Similarly to HORSE Exec Global, an abstraction layer in the interface to the HORSE Exec Global subsystem is included. An explicit Hardware Abstraction Layer is also included to shield design choices for the functionality in the Local Execution. Local Awareness and

Local Configuration modules from technical specifications of connected devices, such as robots and AR devices.

The design and execution modules at both the global and local levels are linked via datastores at the respective levels. At the global level, *process/agent/shopfloor* datastore contains:

- definitions of manufacturing processes, in the form of process models (sequencing of tasks)
- agent models (including capabilities)
- allocation models (role models)
- shop floor models (e.g., 3D models)
- process execution and performance data

At the local level, *task/step/cell* datastore contains:

- task and step model definitions
- contents of tasks, in the form of work instructions/scripts
- task and step execution and performance data

Note that the conceptual data stores containing the above information can be realized with various forms. Physical databases are the most common way to store data, but files (repositories) are also possible (e.g., a BPMN process model is persisted as an XML file).

Part of the functionality of HORSE Design Global and HORSE Exec Global are covered by a process management system. The designed MPMS, as presented in the current chapter, provides most of the functionality of such a process management system. More specifically, the modules of MPMS are mapped to the HORSE system modules. For reasons of clarity, the mapping has been split into two parts, one covering the HORE Design Global (Figure 105) and one covering the HORSE Exec Global (Figure 106) (for a clear and equal mapping, the relevant parts of HORSE Design Global and HORSE Exec Global are considered at the aggregation Level 4). To increase readability, the mapping is also shown in Table 21.

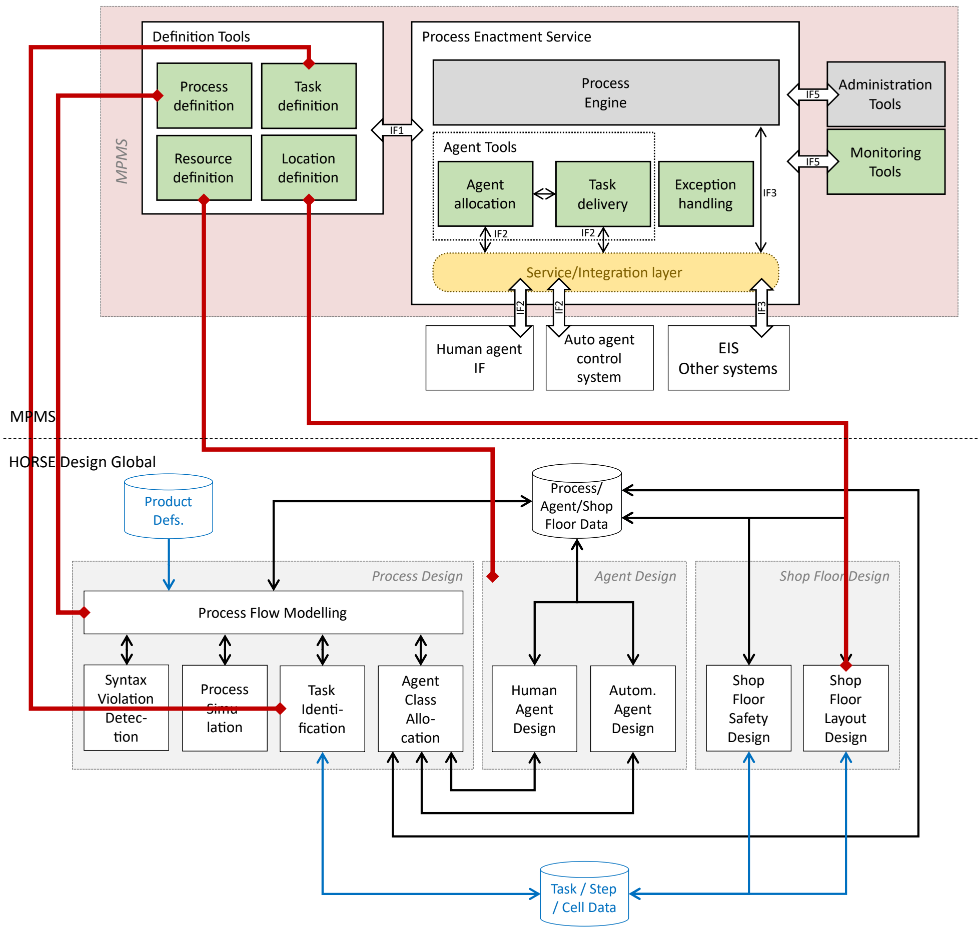


Figure 105: Mapping of MPMS modules to HORSE Design Global modules.

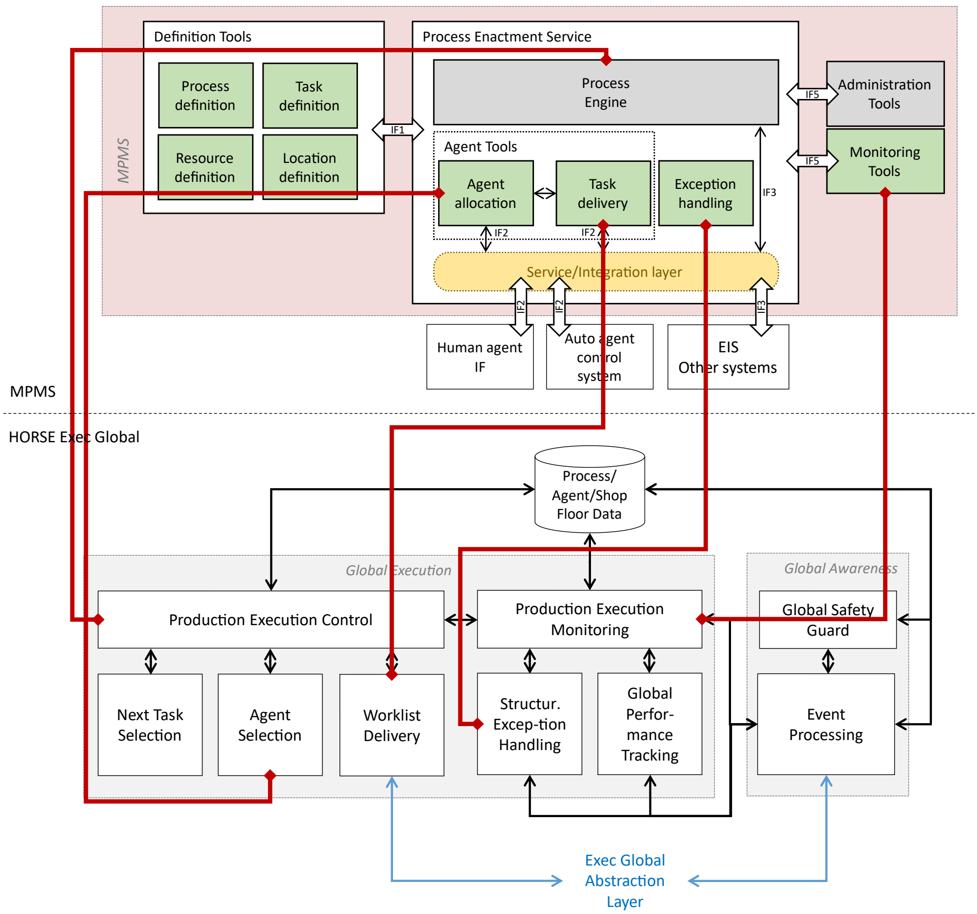


Figure 106: Mapping of MPMS modules to HORSE Exec Global modules.

Table 21: Mapping of MPMS modules to HORSE system modules.

			MPMS modules									
HORSE system modules			**Process Definition**	**Task Definition**	**Resource Definition**	**Location Definition**	**Process engine**	**Task delivery**	**Agent Allocation**	**Exception handling**	**Monitoring Tools**	**Administration Tools**
Design Global	Process Design	Process Flow Modeling	X									
		Syntax Violation Detection										

		Process Simulation										
		Task Identification		X								
		Agent Class Allocation			X							
	Agent Design	Human Agent Design			X							
		Autom. Agent Design			X							
	Shop Floor Design	Shop Floor Safety Design										
		Shop Floor Layout Design				Partially						
Exec Global	Global Execution	Production Execution Control					X					
		Next Task Selection					X					
		Agent Selection							X			
		Worklist Delivery						X				
		Production Execution Monitoring									X	
		Structur. Exception Handling								X		
		Global Performance Tracking									X	

As can be seen, there are modules of the HORSE system that are not (fully) covered by MPMS modules. This, however, does not void the argument that the designed MPMS fits in a CPS architecture for the following reasons:

- Syntax Violation Detection had low priority and was not implemented in the HORSE project. Moreover, the functionality can be supported by an advanced process modeler (conforming to the Process Definition Tools > Validation & Verification module of BPMS-RA) and is considered extra quality feature, without affecting proper process modeling (no design requirement exists for syntax violation per Table 20).
- Process Simulation had low priority and was not implemented in the HORSE project. Moreover, the functionality, while important to ease process modeling and avoid costs, does not hinder proper process modeling and execution (A module for process simulation would conform to the Process Definition Tools > Simulation & Optimization module of BPMS-RA).

- Shop Floor Safety Design is out of scope of MPMS. It is a module to be provided/used by safety engineers.
- Shop Floor Layout Design mostly includes the physical design, but it is the data representation of locations at the shop floor that are relevant for MPMS.

It should also be noted that the HORSE Design Global Task Identification module, which interfaces to Task/Step/Cell data store, contains all the information for defining a task (e.g., input/output parameters, rules, etc.). That means that this module provides the same functionality as the MPMS Task definition module. The information in Task/Step/Cell data store is provided by the HORSE Design Local Task Design module. The naming task design should not be confused with the task definition, as the HORSE module refers to the physical setup/configuration of a task, while the MPMS module refers to the relevant data information.

Regarding interfaces, Table 22 explains how MPMS interfaces provide the desired connections within the HORSE system.

Table 22: Mapping of MPMS interfaces to HORSE system interfaces.

MPMS interfaces	HORSE system interfaces
IF1	Process/Agent/Shop Floor data store is the linking point between design (definition) modules and execution modules. Task/Step/Cell data store does not have a direct link to Global Execution, as MPMS Task definition has with Process Enactment Service (through IF1). This is achieved indirectly through the Task Identification and Process Flow Modeling modules. However, it should not be an issue for Process Enactment Service to communicate with a Task data store.
IF2	All communication with agents is achieved through the Exec Global Abstraction Layer. Section 5.4.2 elaborates more on that aspect.
IF3	HORSE system does not provide explicit integration to other EIS. Connection to a Product Defs. Data store is possible (in which a PLM system typically stores product information) but is left out of scope. Integration through the Exec Global Abstraction Layer can be possible. Section 5.4.2 elaborates more on that aspect.
IF5	Direct interfacing between Production Execution Control and Production Execution Monitor (regarding Monitoring Tools). This can be achieved either through service calls or through querying internal databases in which (execution) data is persisted.

5.4.2 Integration to other systems through middleware technologies

Placing MPMS in the context of a CPS architecture, as the one defined by the HORSE framework, reinforces the proposition of having a central process orchestration system to enable horizontal and vertical integration. Integration requires clear interfacing to other systems. From the designed logical software architecture (Figure 103), IF1 and IF5 are internal interfaces among MPMS modules and, thus, not interesting from systems' integration point of view. IF2 and IF3 are the ones under concern in this section.

MPMS is realized based on existing BPM tooling, as thoroughly discussed in Section 6.3. BPM systems offer various integration options, with the most common:

- REST API[55], to provide access to all relevant interfaces of the engine, e.g., to query for running task instances.
- Connectors, to invoke (business) services on other systems, often based on HTTP and SOAP protocols.
- Data connectors, to push or pull information from data providers such as business intelligence (BI) systems, data lakes or data warehouses.

However, instead of studying such options, a more integrative approach based on middleware technologies is discussed. Middleware is a type of software designed to facilitate the interconnection of a set of software modules (Grefen, 2016). Middleware technologies reduce the number of required interfaces to the number of modules, as module-to-module interfaces are not necessary.

A commonly used type of middleware is a message bus, which relies on the asynchronous exchange of messages between the modules. A message bus middleware has been developed in the HORSE project, as a realization of the abstraction layers, discussed in Section 5.4.2.1.

Another solution of "exchanging" data/information among software modules is with the use of a context broker. A context broker manages context information in a decentralized way[56]. Context information is considered the current state of the surrounding real world, understood as the state of both physical and virtual entities (e.g., a manufacturing task can be considered as a virtual entity). The use of context information helps to develop what is referred to as a "smart factory". Orion Context Broker[57] was the first context broker implementation and is the core and mandatory component of the FIWARE Smart Industry reference architecture[58] (architecture diagram available in Appendix H). The FIWARE platform, with the Orion Context Broker, is selected as a middleware platform in the SHOP4CF project. Integration of MPMS to other systems through a context broker is discussed in Section 5.4.2.2.

Note that from a functional perspective, the message bus and the context broker approaches are different. The former is mainly for delivering information from a sender to one or more recipients, while the latter is to manage information of the current status of the system, which every interested system can access and/or change.

5.4.2.1 Message bus middleware

A message bus-based middleware is designed to deliver messages across clients, using a common message syntax and a central unit that distributes the messages to the receivers. The terms "clients" refers to the senders/receivers of the messages, while the term "broker" is used for the centralized unit processing the messages and performing their forwarding. Figure 107 shows a typical topology of software components based on message bus approach. Clients can be organized into domains, based on functional or physical requirements. For instance, a set of components dedicated for a work cell compose a domain. A dispatcher

[55] REST API (also known as RESTful API) is an application programming interface that conforms to the constraints of REST architectural style and allows for interaction with RESTful web services. REST stands for representational state transfer and was created by computer scientist Roy Fielding - https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm
[56] https://ec.europa.eu/digital-building-blocks/wikis/pages/viewpage.action?pageId=82773700
[57] https://fiware-orion.readthedocs.io/en/master/
[58] https://www.fiware.org/community/smart-industry/

component is used as a mediator among brokers. Per decision, the dispatcher is included within the global components since there is typically only one dispatcher in an entire system.

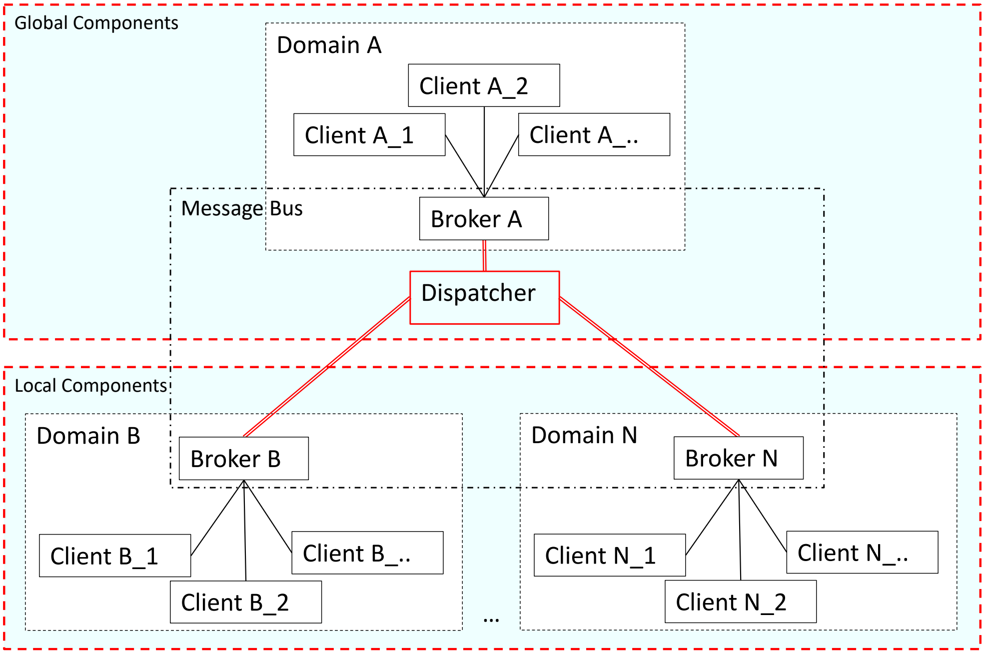


Figure 107: Components topology in a message bus-based middleware approach (from (Arnaudov, 2018b)).

MPMS registers as a client (in a global domain), able to exchange data with other systems. For instance, when a robot control system has registered as a (local) client, MPMS can send task assignment messages. Accordingly, MPMS can receive task status update messages from the robot controller (covering IF2 interface with such communication). Similarly, any other information system can register as a client and, thus, be able to communicate with MPMS (covering IF3 interface).

5.4.2.2 Context Broker

A context broker enables managing the entire lifecycle of context information including updates, queries, registrations, and subscriptions. Various context broker implementations exist (e.g., Orion, Scorpio) and many solutions are based on these (with European Commission supporting these efforts[59]). Regardless the implementation technologies and the data format specifications, context brokers rely on the consumer/producer paradigm.

Figure 108 shows the communication options between context broker and context consumer/producers. A context producer publishes updates of the context information as entities to the Context Broker (IFp). The update can be a change on an attribute value of an entity or a new created entity. For instance, MPMS publishes a new task entity, as an instance of the task concept of the data model. The task entity includes information on involved resources, task input parameters, location, etc. A context consumer works in two modes: 1)

[59] https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/Context+Broker+conformant+solutions

in subscription mode, the consumer first subscribes to certain context changes (IFs). When the context broker makes an update available, the consumer receives a notification about it. In the example above, if a robot control system has subscribed to changes on task entities, it will receive the new task entity created by MPMS. This is the preferred mode of consuming context. 2) in query mode, the consumer queries context broker for specific context information. This mode is not encountered frequently, as it requires active polling to get changes. Instead, with the subscription mode, notifications are received when changes happen. The query mode can be useful when the consumer requires information in a certain situation.

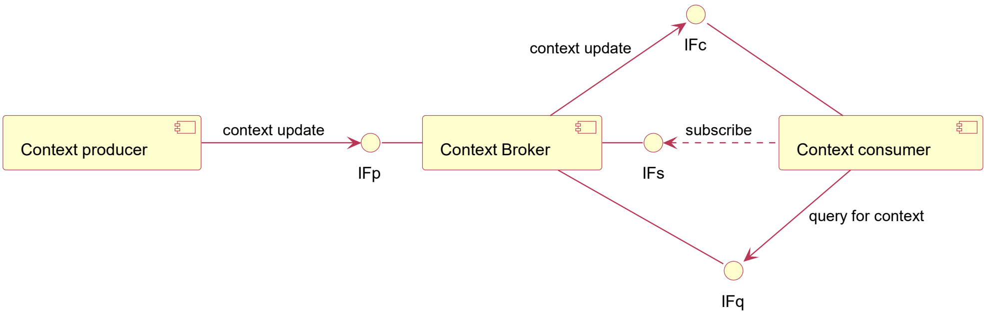


Figure 108: Communication options between Context Broker and Context consumer/producer.

Historical-context consumers exist that query the historical context store that a context broker offers (e.g., for analysis purposes). For reasons of brevity, they are omitted in Figure 108.

Obviously, MPMS can act as either context producer or context consumer. Through context broker, both MPMS IF2 and IF3 are implemented. It is a matter of specifying the right entities (with the right information) that the context broker can manage.

5.4.3 Platform aspect

As integration between MPMS and other systems is important for enabling horizontal and vertical integration, the logical platform aspect of MPMS is discussed in the context of the CPS in which it is part of. Considering the two main integration approaches discussed in Section 5.4.2 (i.e., message bus, context broker), two individual logical platform architectures are presented below. Section 5.4.3.1 discusses the logical platform architecture with message middleware, as developed in the HORSE project. Section 5.4.3.2 discusses the logical platform architecture with context broker, as developed in the SHOP4CF project.

5.4.3.1 Logical platform architecture with message middleware

The HORSE system is first positioned in an enterprise technology landscape, as illustrated in Figure 109. HORSE software components (MPMS one of those) use a cyber-physical middleware to communicate with each other and the hardware components. A database management system (DBMS) is used to manage data in the system. Outside the HORSE system, enterprise information systems communicate through enterprise middleware technology.

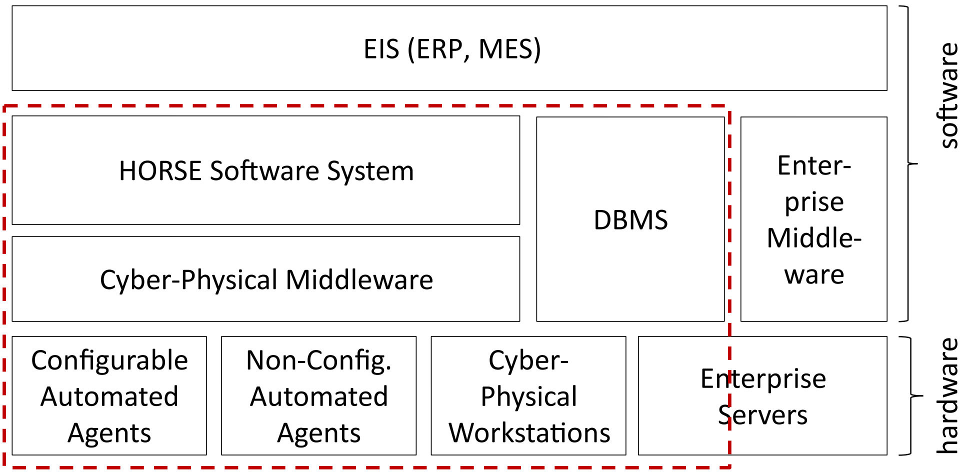


Figure 109: Positioning of HORSE CPS system (red dotted box) in enterprise technology landscape (adapted from (Grefen & Boultadakis, 2021)).

Focusing on the technologies that implement the HORSE system functionality (as designed in the logical software architecture), the technology stack of Figure 110 is designed. The cyber-physical middleware is a websocket-based message bus based on the Open Services Gateway initiative (OSGi) (Pauls et al., 2011) specification. OSGi is a modular system architecture and a service platform that implements a complete and dynamic component model for general module interconnection. It also provides a universal publish-subscribe messaging bus for communication among system modules. The Message Bus is part of proprietary solutions of a technical HORSE project partner, adapted for the needs of the HORSE project. Furthermore, tailored-made OSGi Applications can be used, as software packages that offer powerful and sophisticated component management and interoperability, as well as context-aware assistance of agents (workers, robots) on the production floor in the execution of their tasks. The Hybrid Task Supervisor (part of the Local Execution module of Figure 104) is a type of software to design and synchronize execution steps by agents. Typically, state machines can offer the detailed execution of robotic steps. Robot Operating System (ROS[60]) is a commonly used, open-source, meta-operating system for robots and provides functionality such as hardware abstraction, low-level device control, implementation of commonly used functionality, message-passing between processes, and package management. Open Platform Communications-Unified Architecture (OPC-UA) (Rinaldi, 2016) is used as the interface to advertise and invoke robotic services.

[60] http://wiki.ros.org/ROS/Introduction

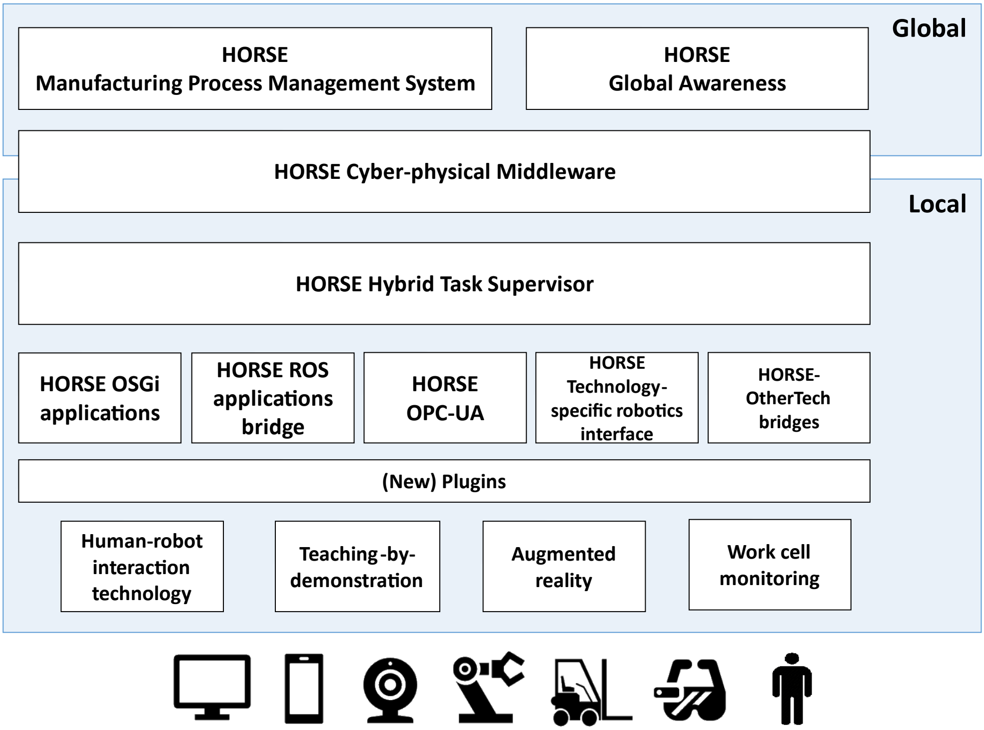


Figure 110: HORSE CPS technology stack.

A more elaborate technology stack of the local technologies is given in Traganos et al. (2021)(omitted here as not the main focus of this thesis).

5.4.3.2 Logical platform architecture with context broker

The high-level logical platform architecture of SHOP4CF system is shown in Figure 111. Vertical adjacency depicts connections between platform components. SHOP4CF components connect to the middleware (including the context broker), but direct connection to IoT devices might be possible. MPMS connects to middleware for better interoperability. Containerization of both SHOP4CF components and middleware is used for easier deployment and control.

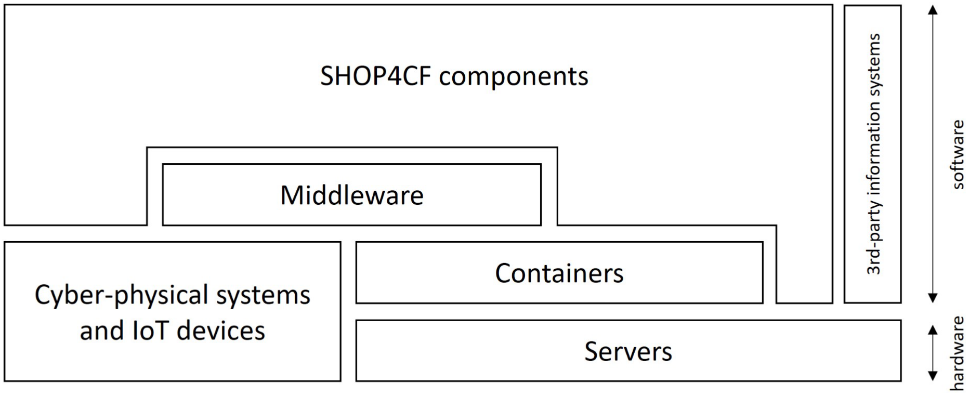


Figure 111: High-level logical platform architecture of SHOP4CF system (Zimniewicz, 2020).

Decomposing the high-level logical platform architecture, a more elaborate view is taken, shown in Figure 112, with the focus on the SHOP4CF components and the middleware layers. As has already been mentioned, the Context Broker is the core component of the FIWARE middleware platform. As it only persists the current state of the environment under concern, historical context store is also provided. Through systems adapters, other enterprise information systems communicate with the Context Broker.

Regarding the SHOP4CF components, five different interoperability classes have been identified to classify them. Each class has different characteristics with different connection points to other systems. A component (e.g., MPMS) can be assigned in more than one class, considering that it provides the essential interfaces.

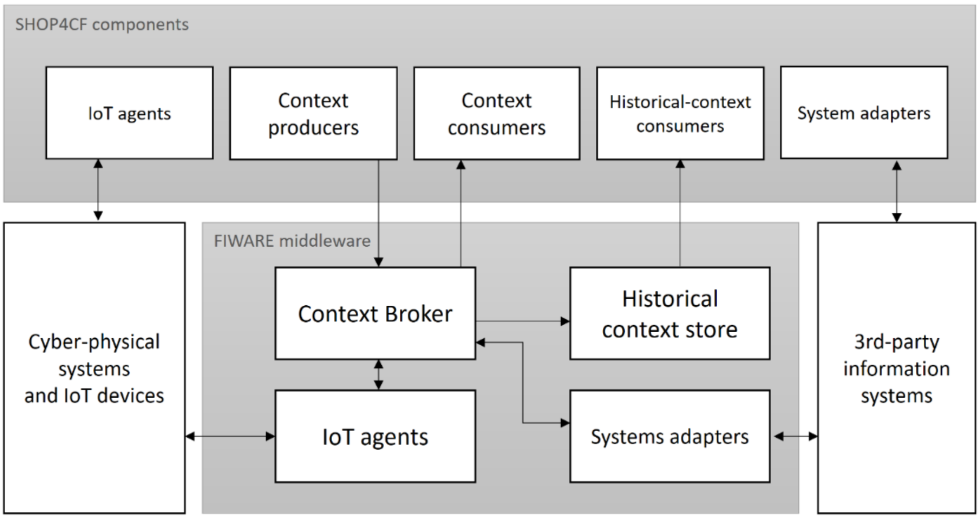


Figure 112: Logical platform architecture of SHOP4CF system, with elaborate view on SHOP4CF components and middleware (Zimniewicz, 2020).

5.5 Chapter conclusion

Given the complexity of processes and the technology heterogeneity in manufacturing environments, a single process management system is proposed for business and manufacturing process orchestration. The system enables horizontal integration with a global control of processes. It also facilitates vertical integration through automated execution of production activities by heterogeneous agents.

This chapter presents the specification of a manufacturing processes management system (MPMS) for process automation and integration in a smart manufacturing context. The specification starts with a logical architecture of the functionality that the system should provide. The architecture is elaborated according to the updated 5-aspect Truijens framework (Grefen, 2016) and is a result of an iterative design cycle process (Wieringa, 2009). As it is based on existing BPM systems, which are typically applied in administrative domains, functionality to be enhanced or completely missing for a smart manufacturing context has been identified and included in the design. New and enhanced functionalities cover the gaps of the studied BPMS reference architectures with respect to application in smart manufacturing. The architecture is complemented with positioning the MPMS in a well-defined architecture of a cyber-physical system for integration of various Industry 4.0 technologies.

The design of MPMS provides guidance on the realization of a unified process management system for operations management. A realization of the system design is presented in Chapter 6, once the operationalization of the conceptual developments of Chapter 3 and Chapter 4 have been included as well. The realization is demonstrated in practical settings and evaluated, discussed in Chapter 7.

CHAPTER 6

Advanced MPMS

The previous chapters have presented conceptual designs that each addresses a specific aspect of the general process complexity issue that this thesis studies. The conceptual designs have to be operationalized to be applicable in specific solutions. Thus, the current chapter first discusses the operationalization of the conceptual designs. The ensemble of the individual operationalizations form an architecture model of an advanced manufacturing process management system. The designed architecture model, combining individual designs, represents the artefact that answers RQ4, i.e., how to support the tackling of the general process complexity issue in smart manufacturing.

Furthermore, a realization of the advanced MPMS architecture model is presented as a proof that design is viable, i.e., the architecture can be realized as an operational system. On purpose, it is referred to as “a” realization and not “the”, as there can exist many operational systems that adhere to the architecture model, based on the selected concrete technologies.

6.1 Chapter outline

The chapter is structured in two main sections based on its two main goals.

Section 6.2 presents the integrated solution for the answering RQ4, once the individual conceptual designs have been operationalized. The operationalization of modeling constructs (designed in Chapter 3) and the operationalization of exception handling (designed in Chapter 4) are executed in the context of BPM theories and technologies (e.g., using the BPMN 2.0 specification). These are discussed in Section 6.2.1 and Section 6.2.2 respectively. Section 6.2.3 presents the development view (of K4+1 framework) of the MPMS specification, whose logical view has been presented in Chapter 5. The resulting artefact for RQ4 is presented in Section 6.2.4, with a focus on the software architecture.

Section 6.3 presents an operational system as a realization of the designed artefact. The realization is based on selected concrete technologies. As MPMS builds on existing BPM tooling, an available BPMS platform is first presented in Section 6.3.1. Section 6.3.2 showcases the application components that realized the software functional components of the architecture. Section 6.3.3 presents a deployment diagram of the system.

The chapter outline is illustrated in Figure 113, following the DSR approach (i.e., it is an elaboration of the high-level DSR diagram of Figure 13 with respect to the part referring to the design of the overall solution). The chapter concludes in Section 6.4.

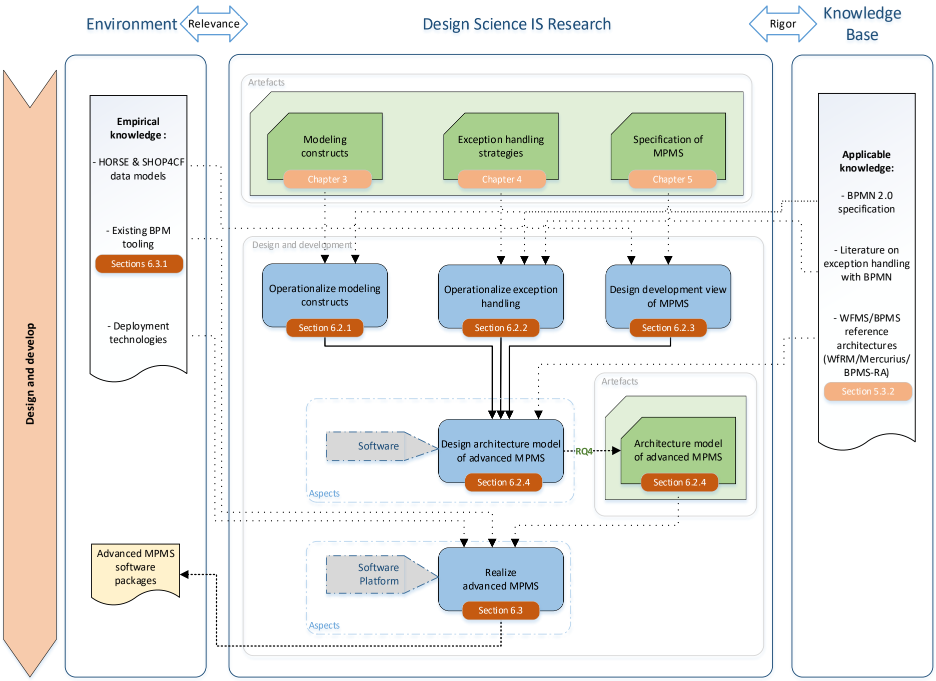


Figure 113: DSR approach for RQ4 - Design of an advanced Manufacturing Process Management System for tackling process complexity. A system realization is developed as well.

6.2 Integrated solution

The integrated solution, i.e., the ensemble of individual designed artefacts, constitutes an architecture model that acts as a blueprint for realizing a process management system for manufacturing operations. The specification of such a system has been conceptually designed (i.e., from a logical point of view) in Chapter 5. It remains to discuss the development view. To claim the "advanced" characterization, it shall include the functionality that has been conceptually designed in Chapter 3 (modeling patterns) and Chapter 4 (exception handling). These designs shall also be operationalized first. Thus, this section presents the operationalization of the modeling constructs in Section 6.2.1, the operationalization of exception handling in Section 6.2.2, and the development view of MPM in Section 6.2.3. Then, Section 6.2.4 presents the final designed artefact.

Similarly to the design of MPMS specification (Chapter 5), the operationalizations of the individual conceptual designs have been performed in design cycles (Wieringa, 2009). For instance, the BPMN process model for the synchronization mechanism has seen various iterations to reach its final form, presented in Section 6.2.1.2. Initially, before the recipe notion had been introduced, a controller to buffer process instances was implemented. That controller, though, was not taking into account the multi-instance perspective and thus had to be revised, leading to the idea of recipes. Accordingly, the exception handling mechanisms had to be revised to cover the final designed list of the exception handling strategies.

6.2.1 Operationalization of modeling constructs

Chapter 3 presents the design of a few modeling constructs to represent concepts encountered in manufacturing processes, and with the goal to ease their executability. Some of those can be directly applied/configured during process modeling, according to the BPMN 2.0 specification[61]. More specifically, the task repetitions patterns (Section 3.3.2), the task queue management construct (Section 3.3.3), the modeling of collaborative assembly process (Section 3.4.2), and the deferred task parallelism constructs (Section 3.4.3). Process modelers have to adapt those constructs per need (e.g., create as many branches as the number of the task types on the task queue management construct), or configure the relevant parameters (e.g., update the condition variables on the deferred task parallelism construct). Consequently, these constructs do not require specific operationalization (at a generic level). On the other hand, the task delivery patterns (Section 3.3.1) and the synchronization mechanism (Section 3.5) require more elaboration to be operationalized by a realized system. These are discussed in the following two subsections respectively.

6.2.1.1 Operationalization of task delivery patterns

Delivering a task directly to an automated agent, or in a module that handles tasks for agents (be it a robot controller or a module to present human tasks in a custom UI) is split into three main phases, namely starting, executing, and ending (as has been shown in Figure 57). The "delivery" concept should not be confused with the starting phase, as the focus here is not only to send task instructions to an agent, but also wait for task's completion. That means that all patterns should achieve the "waiting" state, i.e., while the agent actually performs the work.

The "Send and Receive Tasks" pattern (see Figure 57) achieves the waiting state in a straightforward way. The Send Task is configured to send any information regarding the task (e.g., task input parameters), through a service call. The process engine that handles the automation of the process models advances the workflow to the Receive Task. An external call (from the auto agent/external module), when the work is completed, triggers the Receive Task and the process instance can move to the next element.

For the other two patterns, this decoupling of "sending – waiting – receiving" is not that straightforward. Let us consider first the Service Task pattern (see Figure 57) to explain a few concepts. Figure 114 illustrates the sequence of messages of a synchronous service call, as typically configured by a Service Task. The synchronous service call follows the well-known request/response design pattern[62]. The process engine performs a service call and waits for a response by blocking the transaction thread. Once the response is received, the thread is unblocked.

[61] https://www.omg.org/spec/BPMN/2.0/

[62] http://www.servicedesignpatterns.com/ClientServiceInteractions/RequestResponse

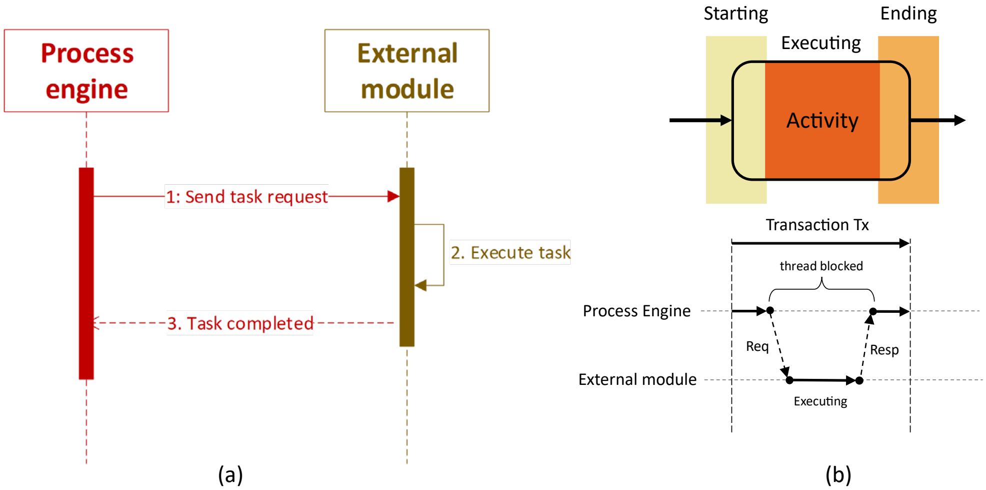


Figure 114: Synchronous service call through a Service Task, as (a) UML sequence diagram, (b) graphical representation.

The asynchronous service call follows the request/acknowledge/callback pattern[63]. Figure 115 illustrates an example of the desired task delivery with asynchronous service call. Once the external module receives a task request, it puts it in a queue (assuming that it can handle the queue). Then, two options are available. Either the queue handler forwards it to the agent (service processor), or the agent requests it (e.g., through a polling mechanism). The agent performs the work and finally sends a callback message to the process engine. The callback itself can be sent either synchronously or asynchronously. The important point here is that the transaction thread is not blocked.

[63] http://www.servicedesignpatterns.com/ClientServiceInteractions/RequestAcknowledge

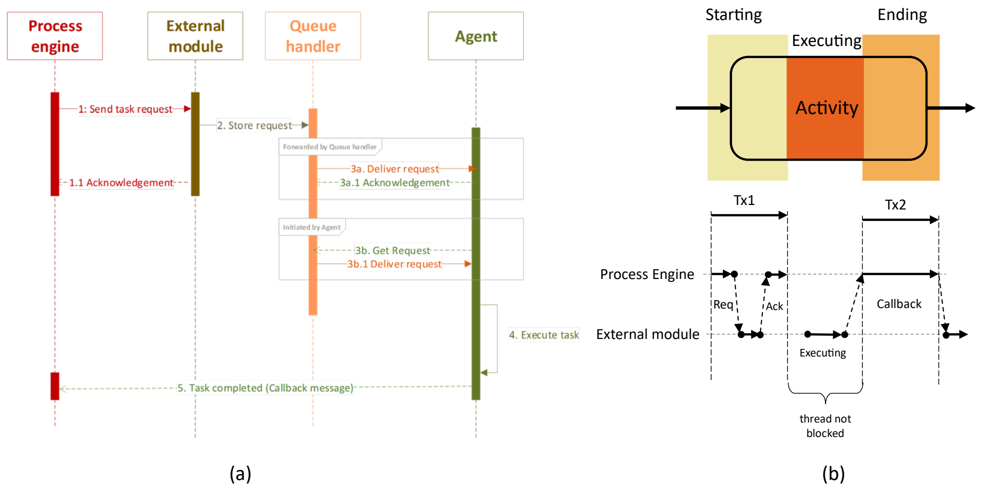


Figure 115: Asynchronous service call, desired for task delivery, as (a) UML sequence diagram, (b) graphical representation.

The same functionality shall be achieved through the use of the User Task (or any new task type defined as an extension to the BPMN specification). Some kind of delegate code shall be sent to an external module upon creation of the task (starting phase). Upon task completion, the callback message shall be received by the process engine.

Regarding the reception of the callback message, three main options are possible:

- The process engine provides a REST API endpoint which can be directly invoked to complete a task. This option gives control to the external module to handle the completion of tasks.
- An application module embedded to the process engine provides a controller mechanism (e.g., endpoint) to receive callback messages. Handling those messages means that the application can interact with the task instances, e.g., calling process engine's internal API to complete a task instance.
- The process engine receives messages through a non-interrupting event subprocess. Handling those messages can be easier as the correlation to the running task instances is more explicit.

The second option is discussed in Section 6.2.3, which elaborates on the integration options to other systems. The third option, essentially, provides the same functionality as the second option, but it is a more explicit option from a process modeling perspective. The third option is illustrated in Figure 116. Note that the entire process represents the called subprocess of Figure 54. It is a simplified extension of Figure 55, from which the allocation paths have been excluded for sake of simplicity. The pattern of Figure 116 is not only relevant for receiving final task status message (e.g., "task completed" or "task failed"), but it can be applied for handling any task messages. For instance, it can be used for handling messages of the actual initiation of the task, linking to the pattern of Figure 64.

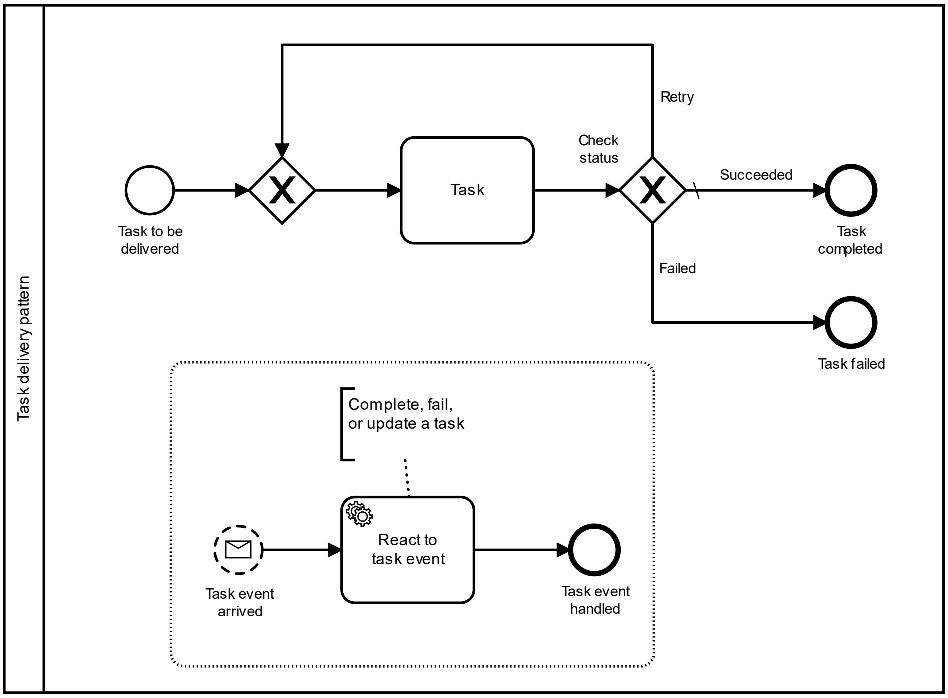


Figure 116: Pattern for receiving task events.

6.2.1.2 Operationalization of synchronization mechanism

The operationalization of the recipe system (Section 3.5.3) consists of three main parts:

i. A data model to represent the formal entities.
ii. A software package to represent the data entities as objects and to operationalize the formal algorithms.
iii. A BPMN 2.0 model to represent the Recipe Controller.

The data model provides all details on attributes of classes and relationships between classes. It is available in Appendix I. The software package follows the object-oriented programming paradigm, written in the Java programming language. It contains classes to represent the data entities, to handle the messages, and to evaluate recipes according to the mapping and pool algorithms. Details of accessing the source code can be found on Appendix J.

The process model of the Recipe Controller is shown in Figure 117. Two main process models are also illustrated in Figure 117 to ease the elaboration of the controller's functionality. Four types of messages are included in the recipe system:

- Submit. Used by a process model to submit a process instance to a pool.
- Cancel. Used by a process model to remove a process instance from a pool before release through fulfilment.
- Release. Used by the recipe controller to indicate that the process instance(s) of the pool(s) that fulfil the recipe shall continue their flow (currently waiting at a Release Receive message event).

- Start. Used by the recipe system in a similar way as the Release message type, but to start a new process instance (according to the recipe), instead of continued the flow of a running instance.

In short, process models that include synchronization points and are part of a recipe submit (accordingly cancel) their process instances in the recipe controller (persisted in the corresponding pool(s) according to the recipe definition). The recipe controller evaluates the recipes and upon fulfilment of a recipe releases the continuation of the flow of the relevant process instances (or starts a new process instance).

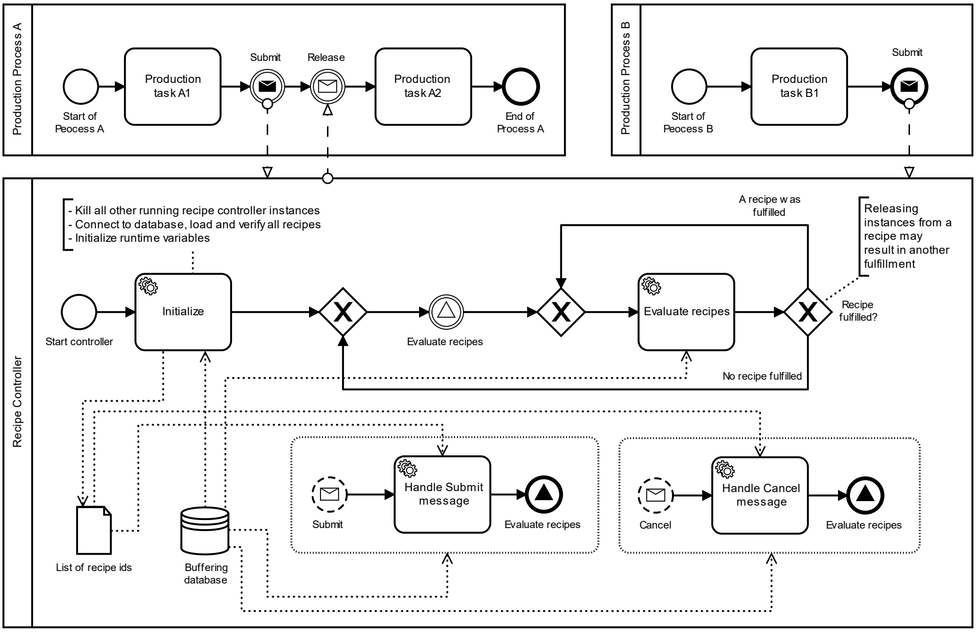


Figure 117: The Recipe Controller implemented as BPMN 2.0 process model.

6.2.2 Operationalization of exception handling

A list of exception handling strategies/patterns (Table 17) is identified in Section 4.3.3.1 for dealing with deviating behaviors during manufacturing operations. Based on the type of occurred exceptions and considering also the critical MOM KPI(s), a set of guidelines in the form of decision trees are designed to select a suitable strategy/pattern (Section 4.3.3.2). Regardless, though, of how an exception handling strategy/pattern is selected, there should be support by a process management system, both from a modeling and an execution perspective. Thus, this section presents BPMN 2.0 modeling constructs to support the selected exception handling strategies/patterns.

Table 23 discusses how support is provided for the exception handling strategies/patterns of Table 17 through the BPMN constructs of Figure 118 and Figure 119, shown below. Figure 118 presents a construct for task delivery and exception management. The construct shall be used as a subprocess to be called by main (production) processes. The construct is an extension of Figure 116, which has been used as a simplified version to illustrate the task delivery to agents. It is also an extension of Figure 55 – that figure focuses on the allocation mechanism, without exception handling. For reasons of brevity, allocation is simplified in Figure 118, by using only a single task (type). This does not affect the exception handling as exceptions raised/indicated by either a human agent or an auto agent are covered, as discussed in Table 23. Figure 119 presents two auxiliary processes to handle exceptions at a process level. It should be noted that functionality to register an exception (e.g., through a Service Task) is omitted for reasons of brevity.

Existing work has inspired the design of the BPMN 2.0 modeling constructs. For instance, constructs for immediate or deferred fixing have been proposed by Lerner et al. (2010) and Reichert & Weber (2012b). Ritter & Sosulski (2014, 2016) have also designed BPMN constructs for exception handling. However, those works do not cover the entire set of handling strategies that the current research has rigorously identified. Moreover, the constructs presented in this research form a compact and more comprehensive way to encapsulate exception handling functionality. This is achieved through: i) one subprocess that can be called from any activity of a main process (Figure 118), compared to patterns that have to be iterated over any task in a process model and thus leading to expanded process models that can affect readability and understandability, ii) two auxiliary processes (Figure 119) for centrally managing exceptions at process level, modeled outside the scope of the main processes in concern instead of modeling them inside the processes in concern.

Table 23: BPMN support of exception handling strategies/patterns (per Table 17).

Exception handling strategies/patterns		BPMN support explanation
1. Retry	a. Restart	A human agent gets the option to select "Retry" in a current (User) task. An automated agent sends a message, caught with the "Task event arrived" message event of the "Task Events Listener" event-subprocess, with the "Retry" value passed. Upon receiving such message, the system shall advance the workflow into the "Retry" path.
	b. Rework	Achieved through the "Retry" path, as in Restart approach.
	c. Immediate fixing	When immediate fixing is required, typically to be performed by a human operator, an error is raised and the "Fix" task is assigned to a human agent. The operator indicates then the new status and the workflow continues.
	d. Deferred fixing	When deferred fixing is required, assigned agents are released (i.e., considered as not occupied). Task information is preserved in a pool that controls tasks to be performed at a later stage. The handling of such tasks is controlled by the "Deferred Task Controller" process, which continuously monitors for available "deferred" tasks. The "Handle deferred tasks" Service Task of the "Deferred Task Controller" includes the business logic on when to execute a deferred task, implemented per scenario/need.
2. Try alternatives	a. Replace order	When a request for replacing an order arrives, caught by the "Process event arrived" message of the "Process Events Listener" event subprocess of the "Event Handling Process", related running instances are canceled, so new ones can start for the new order. The cancelling of related instances is performed based on order information (e.g., OrderNo). That means that

		the corresponding (production) processes shall have correlated order information with running instances.
	b. Change/Move to new settings	This approach typically involves many activities besides execution of the regular production processes. For instance, checking whether a spare production line is available for starting a new production run, or whether new setups have to be implemented before moving. This goes beyond the scope of the operational level (i.e., decisions on a tactical or even strategical level) and not always covered by standard modeling patterns. In case the changes/moving can be performed easily, the cancellation of related running instances, as performed for 2.a) can be applied.
3. Compensate	a. Add/Insert tasks	Performing an additional task, typically performed by a human operator, is an ad hoc way to compensate some work done in case of errors. The additional work can vary and often depends on the experience of the operator to resolve an issue. What is important for the system is the confirmation that the additional work is completed (maybe with providing some input information), as captured by the "Perform additional task" User Task.
4. Rollback	a. Jump backwards	In case no additional task is needed (covered by 3.a)), a compensation trigger is raised from the "Task delivery and exception management pattern". It is then caught by the parent task in the main process, triggering any compensation actions (through a Compensation boundary event).
5. Suspend/Resume	a. Delay the activity	In case of a task to be performed by a team involving a human agent, the delay is provided as an option through the "Execute delayed task" User Task. This is basically a way to put the workflow in a waiting state and getting the confirmation of the human agent on when he plans to start working on it. In case of a task to be performed by auto agent(s), any delayed execution is typically handled through queueing mechanisms as discussed in Section 3.3.3. Of course, queuing mechanism can be applied for tasks delivered to human agents as well but given the flexibility of operators to perform/continue delayed work, direct support is provided.
	b. Suspend/Resume case	Upon reception of an event, caught by the "Process event arrived" message of the "Process Events Listener" event subprocess of the "Event Handling Process", requesting for suspending a case, a Service Task with the required business logic suspends the related running instance(s) of that case (e.g., linked to an order). Similarly, a suspended case can be resumed with a Service Task.
	c. Suspend/Resume all cases	Similarly as in 5.b), more than one cases can be suspended/resumed.
6. Continue process	a. Skip	Both human and auto agents are given the option to skip an assigned task (whether they started working on it or not). Human agents can select the option through a tasklist application, auto agents can pass this information on task messages.
	b. Jump forward	In case an exception during an activity requires to jump forward, there should be indication on which should be the next task. The skipped tasks are marked, so they are not executed (as handled by the "Skipped" XOR gateway.

	c. Ignore	Exceptions are ignored, no action is needed (except registering the exception for analysis purposes).
	d. Do nothing	No action is needed.
7. Terminate	a. Force complete activity	Both human and auto agents are given the option to complete a task (whether its actual work is finished or not). Human agents can select the option through a tasklist application, auto agents can pass this information on task messages.
	b. Force complete case/process	Upon reception of an event, caught by the "Process event arrived" message of the "Process Events Listener" event subprocess of the "Event Handling Process", requesting for force completion of a case, a Service Task with the required business logic completes the related running instance(s) of that case (e.g., linked to an order).
	c. Force fail activity	Similarly as in 7a), agents are given the option to fail a task.
	d. Force fail case/process	Similarly to 7.b), running instance(s) can be forced failed.
	e. Discard workpiece	This does not require any action/support (except registering the exception and the discarded workpiece for analysis purposes).
8. Escalate	a. Reallocate	In case an exception during an activity requires a reallocation (e.g., low battery of an AGV), the reallocation mechanism is called again, i.e., Exception boundary event looping back to "Select Team" Decision task.
	b. Terminate process (7) and propagate	In case an exception during an activity requires termination of a case/process, message is sent to the "Event Handling Process". The latter process takes care to propagate the exception message to upper management.
	c. Contact/Coordinate	In case of an exception failure, an "Out of normal action" task is triggered where the agents contact/coordinate with other roles no whether and how issues can be resolved.

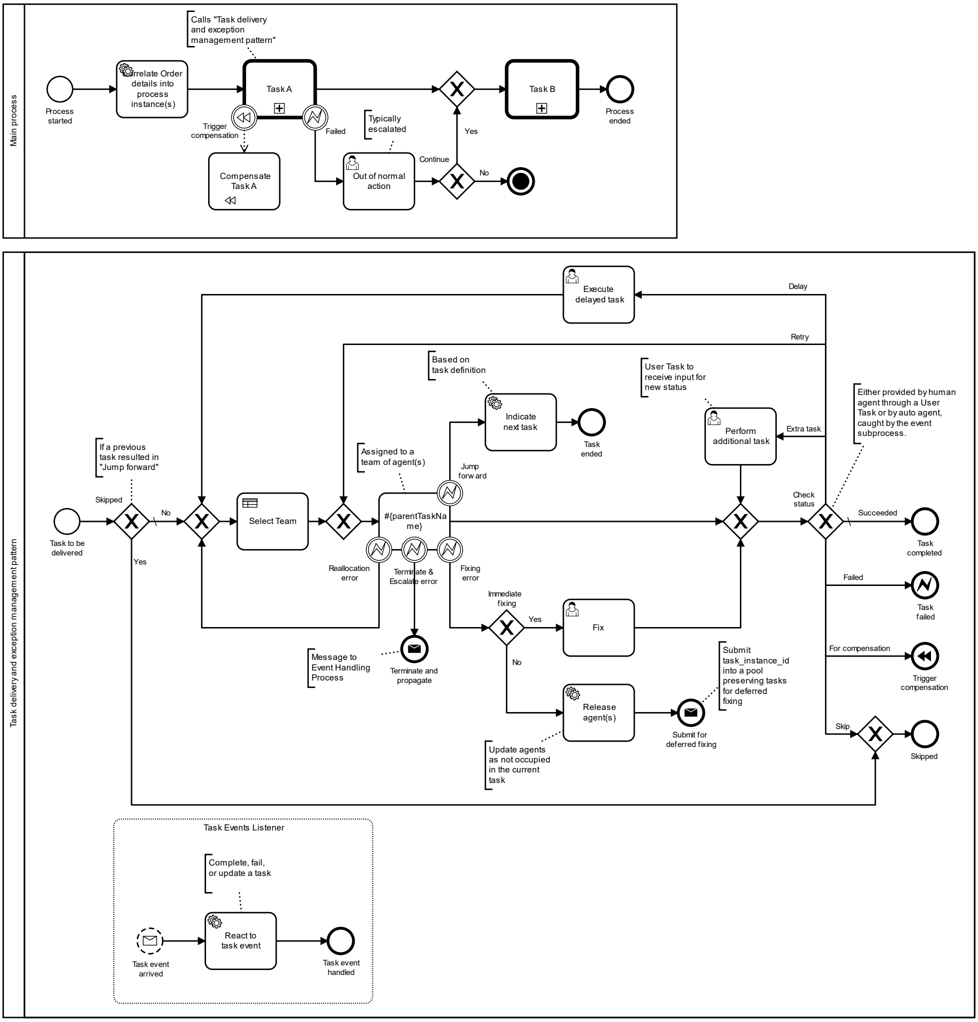


Figure 118: BPMN model construct for task exception handling. The task construct (bottom process) is a subprocess to be called by tasks modelled in any main process (top process).

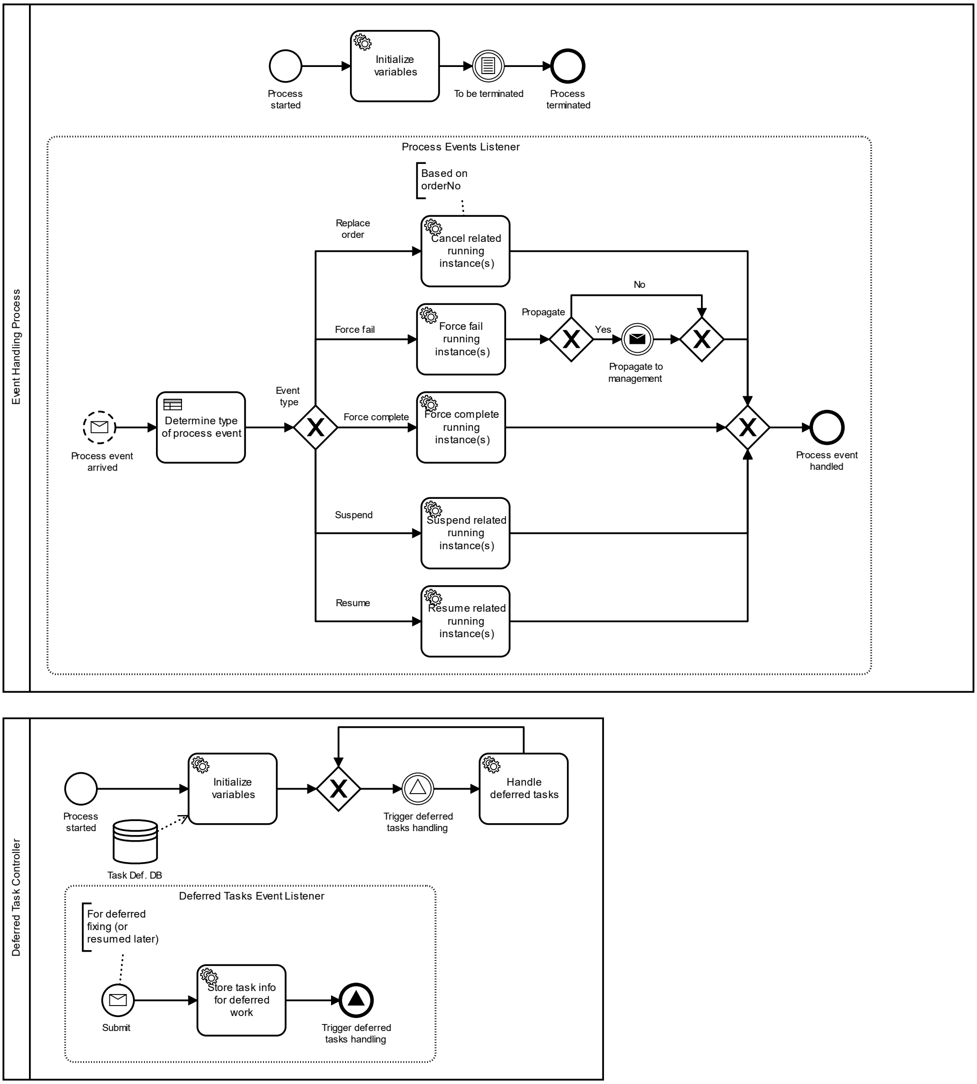


Figure 119: Auxiliary processes for exception handling: (top) process event handling, (bottom) handler for tasks to be executed at a later stage (e.g., for deferred fixing).

The “Determine type of process event” Decision task in the “Process Events Listener” event subprocess of “Event Handling Process” (top process of Figure 119) encapsulates the logic to distinguish what kind of event has been captured and what support is required. Events captured through the “Process event arrived” message of the “Process Events Listener” event subprocess of the “Event Handling Process” shall contain the right information to make the decision. To structure required information, a UML class diagram is designed, shown in Figure 120. The diagram is an operationalization of the event concept model (Figure 99) of the data aspect of MPMS (Section 5.3.3.3).

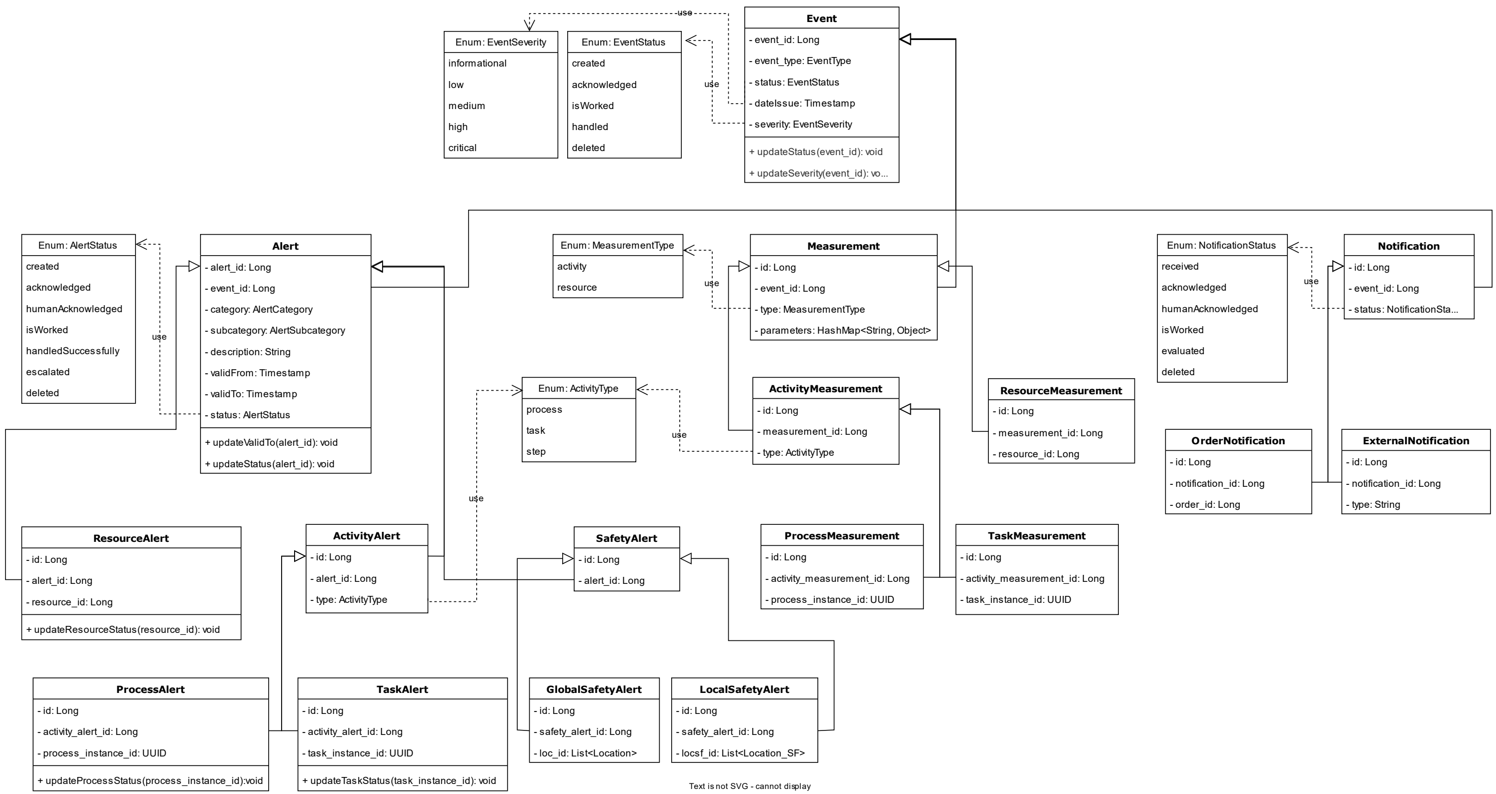


Figure 120: UML class diagram as operationalization of Event concept model of Figure 99.

6.2.3 Development view of MPMS specification

The development view of K4+1 framework (see Figure 81) discusses the operationalization of the logical view of the MPMS specification, discussed in Chapter 5. The logical view touches all five aspects of the UT5 framework (Figure 82), discussed in Section 5.3.2 and Section 5.4.3. Though, not all aspects are relevant from an operational point of view.

The organization aspect (Section 5.3.3.1) analyzes the physical hierarchy of a manufacturing enterprise and its structure has been considered in the location concept model (Figure 100) of the data aspect (Section 5.3.3.3). Similarly, the process aspect (Section 5.3.3.2), which discusses the hierarchy and logical sequence of activities, has been considered in the activity concept model (Figure 96 and Figure 97) of the data aspect. The integrated concept model (Figure 101) of the data aspect shall be operationalized. The operationalization of the software aspect (Section 5.3.3.4) is better discussed through concrete details, i.e., through a realization system (Section 6.3). Though, the operationalization of the external interfaces (IF2/IF3) of the logical software architecture (Figure 103) deserves detailed elaboration regardless concrete technologies of a realized system. Finally, the operational view of the platform aspect (i.e., concrete deployment details) is better discussed on realized system(s) (Section 6.3) and through the integration and demonstration of the system in real-world environments (Section 7.1).

Thus, the current section presents the operationalization of the concept model of the data aspect in Section 6.2.3.1, and the operationalization of the external interfaces of the system in Section 6.2.3.2. Deployment details of a realized system (physical view of K4+1) are discussed in Section 6.3. This discussion refers to the realization of MPMS as a standalone system. Deployment details of a CPS in which MPMS is embedded are discussed in Section 7.1.

The above explanation is also illustrated in Figure 121. The link between the data and software aspects is kept to highlight that the operational views of each aspect should consider details from the other aspect. For instance, the technical data model (data aspect) should include details of software module(s) (software aspect), e.g., data attributes of the process engine.

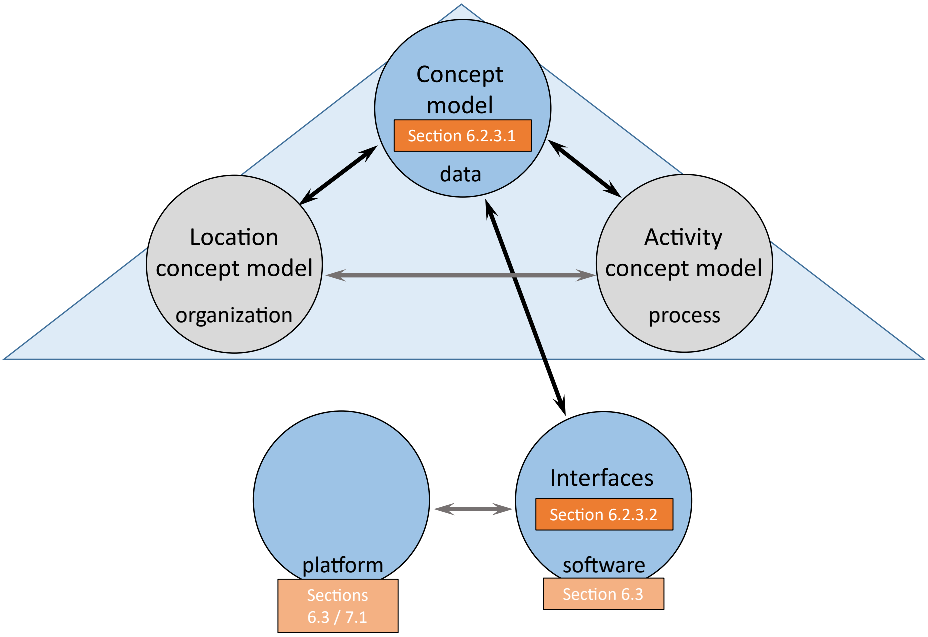


Figure 121: UT5 aspects with logical design(s) from Chapter 5 that shall be operationalized, mapped to corresponding sections.

6.2.3.1 Operationalization of concept model (data aspect)

Section 5.3.3.3 discusses relevant concepts to represent manufacturing entities. Different concept models are designed and linked into an integrated concept model (Figure 101). This high-level concept model is operationalized by detailing the attributes of each concept. Section 6.2.2 already presents the data model with respect to the event concept model. This is also included in Figure 122, which covers all concepts.

A few design decision/explanation points with respect to the data model of Figure 122:

<u>*Design decision 1):*</u> While the Location class represents the equipment hierarchy in terms of structure, a more detailed representation of location points is necessary, captured by the Location_SF (shopfloor) class. Examples of entities in this class are the exact location of devices or storage bins on the shopfloor layout.

<u>*Design decision 2):*</u> A location point (from Location_SF class) is described with respect to its functional purpose as source or target. This is important for transportation tasks (e.g., an AGV to move from point storage A to station B). Apparently, the same location can function either way, depending on the context of the activities taking place.

<u>*Design decision 3):*</u> The Step class (from Figure 101) are omitted as they are more relevant for modules of the local control level (and thus to avoid cluttering the diagram).

<u>Design decision 4):</u> The entire Team class and a few attributes in ProcessDef, TaskDef and AgentSpec classes are grayed, as representative examples of attributes used for the allocation mechanism. More elaborate attributes and relationships are covered in Erasmus (2019).

<u>Design decision 5):</u> The Agent class is not further specialized in Human and Auto agent classes, to avoid cluttering the diagram. A technical implementation of the data model (e.g., in a physical database) should take them into account, as each have different characteristics (Erasmus, 2019).

<u>Design decision 6):</u> Statuses values in all enumerators can be adapted/extended, but have to be respected by all components that will use the data model, regardless whether they refer to global or local level.

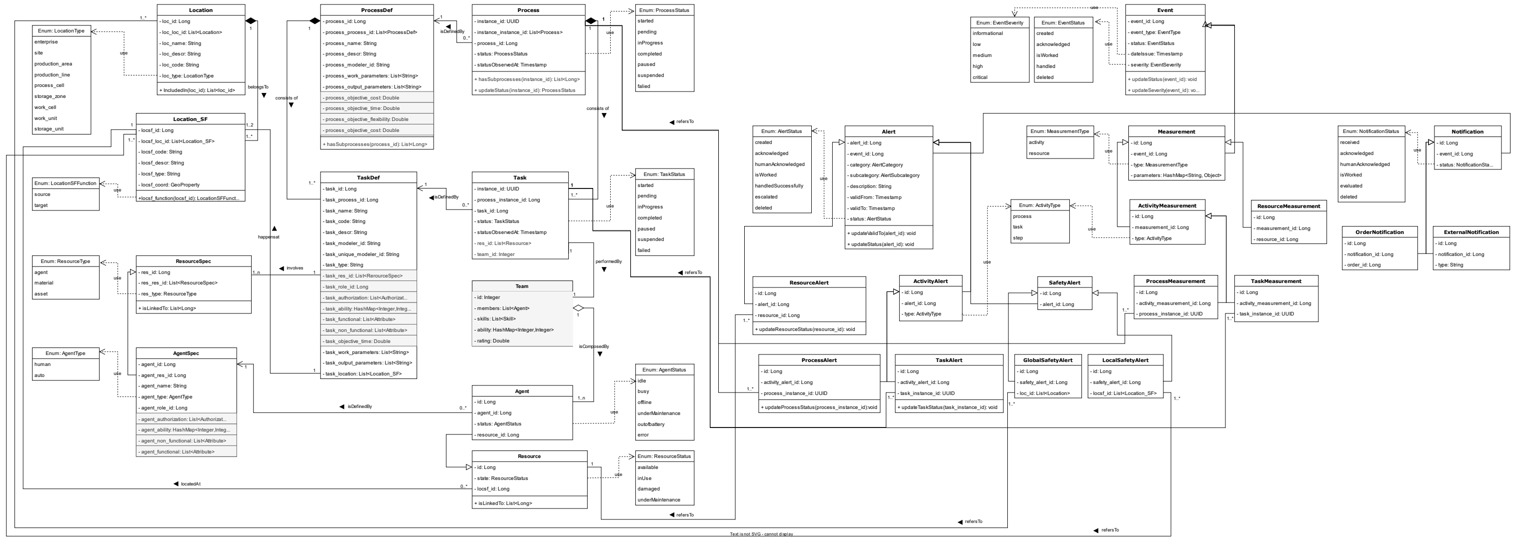


Figure 122: UML class diagram as operationalization of the integrated concept model of Figure 101.

6.2.3.2 Operationalization of interfaces (software aspect)

The design of interfacing MPMS to other systems (Section 5.4.2) includes two main options, integration through message bus middleware and through context broker. Thus, the operationalization of these two options is discussed separately below.

6.2.3.2.1 Interfacing to Message bus middleware

Interfacing to a message bus requires first to register MPMS as a client (per Figure 107). An application module can handle the registration and creation of endpoints for handling messages between MPMS and the message bus. Such an application has been operationalized as part of the current research. Details of accessing the source code can be found on Appendix J.

To ease the integration of MPMS to the message bus, two auxiliary BPMN processes are created, as shown in Figure 123. Before any production process runs, the MPMS Handling process (top swim pool of Figure 123) takes care to register MPMS as a message bus client. A number of options are given to MPMS administrators/users to make requests to the message bus, modeled in the bottom swim pool of Figure 123. These include checking for active connection(s), getting list of registered clients (the names of which are used to send messages to specific recipients), informing main processes for active registration, and disconnecting MPMS (e.g., for maintenance or troubleshooting purposes).

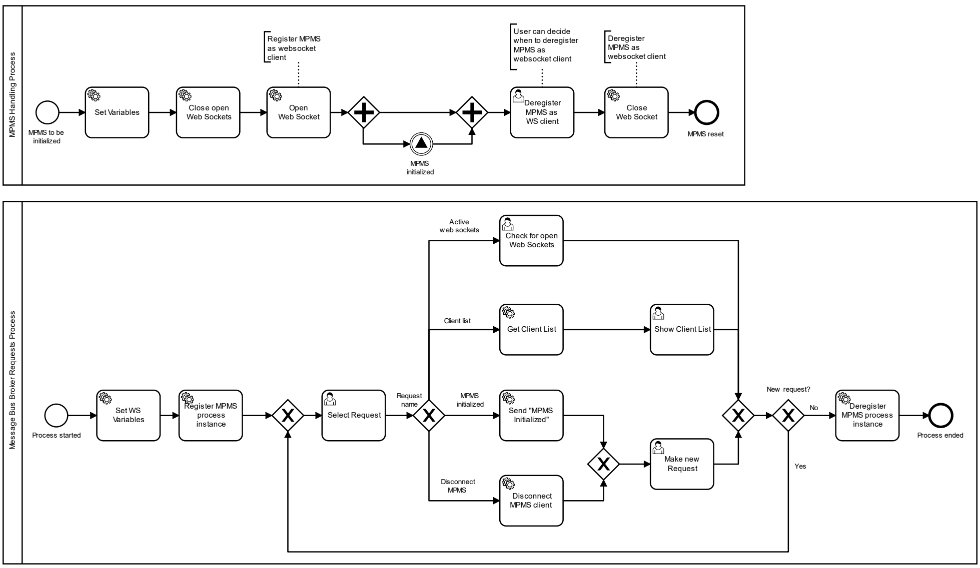


Figure 123: Auxiliary BPMN processes to handle MPMS integration to Message bus middleware.

Once registration of MPMS as a client is handled, the main production processes can start, in which messages to/from the message bus are exchanged. Figure 124 illustrates an example. The top swim pool refers to a high-level view of the process. In the beginning, a check is performed on whether an active connection to the message bus has been established, otherwise the process is paused (intermediate signal catch event) until registration has been established. Relevant information is received from data stores (e.g., product info, or task parameters). Then, a subprocess is called to actually execute the production activities (bottom swim pool). When a task requires to be delivered to an agent through the message bus, a task assignment message is sent upon task instance creation. When the actual work has finished, a task completion message is forwarded by the agent to MPMS. The application that handles the messages completes the task through process engine's API (or any specification).

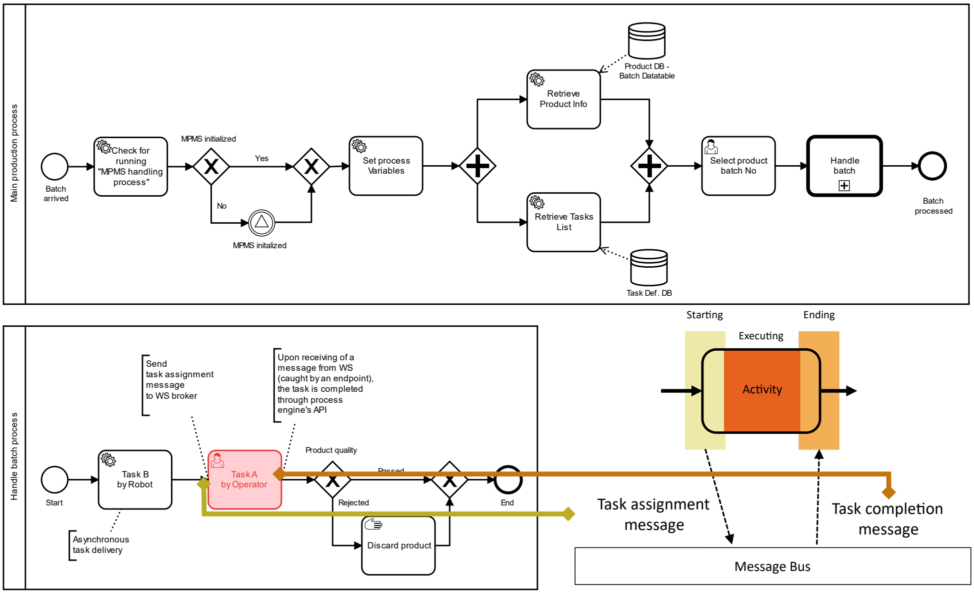


Figure 124: Example of production processes in which interfacing to message bus is required.

Example messages are shown in Figure 125, expressed in JSON[64] format. The syntax of the messages is specified by the message bus implementation (e.g., the HORSE message bus specification (Arnaudov, 2018a)). Similar messages can be constructed to exchange information on processes, events, agents, etc., respecting the data models (Section 6.2.3.1).

```
{
 'Topic': 'task_assigned',
 'SenderID': 'mpms',
 'Receivers': 'rosbridge',
 'Subtype': 'notification',
 'Timestamp': '20180117103348',
 'ExternalBrokers': '*',
 'SenderBroker': '',
 'Body':{
         'args':{'agent_ids': '1',
                 'task_id': '4',
                'task_instance_id': '7d69-5c05',
                'process_instance_id': '81dd-11e3'
               },
         'in_variables':{'batch_no': '34'}
        }
}
```

(a)

```
{
 'Topic': 'task_completed',
 'SenderID': 'rosbridge',
 'Receivers': 'mpms',
 'Subtype': 'notification',
 'Timestamp': '20180117103348',
 'ExternalBrokers': '*',
 'SenderBroker': '',
 'Body':{
         'out_variables':{'quality_result': 'passed'}
        }
}
```

(b)

Figure 125: Example messages for task delivery through message bus, expresses in JSON format (a) task assignment, (b) task completion.

[64] https://www.json.org/json-en.html

Note that other type of messages can be exchanged as well, e.g., alerts, not only task-related information.

6.2.3.2.2 Interfacing to Context Broker

An application module can handle the subscriptions, the context update and the context consuming of MPMS in relation to a Context Broker (per Figure 108). As a context consumer, MPMS, shall subscribe to specific entities that are relevant for receiving context. These entities are: process entities (to receive changes on process level), task entities (to receive task status changes), resource entities (to receive changes on availability), and alert entities (to receive new alerts). The entities follow a specification[65] respecting the data models (Section 6.2.3.1). As context producer, MPMS shall post entities that are relevant for other components to consume. For instance, task entities to be delivered to a robot controller module. Such an application has been operationalized as part of the current research. Details of accessing the source code can be found on Appendix J.

To ease the integration of MPMS to a Context Broker, an auxiliary BPMN process is created, as shown in Figure 126. The process automates the handling of subscriptions creation. First, all existing subscriptions are retrieved, then a check is performed on whether desired subscriptions already exist. If not, new subscriptions are created.

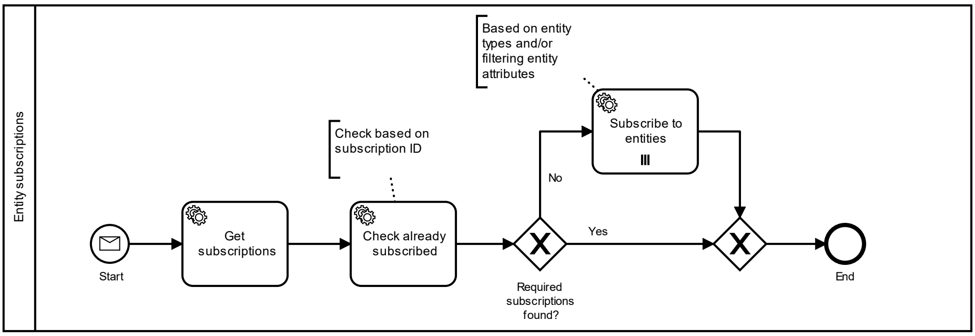


Figure 126: Auxiliary BPMN process to handle subscriptions to Context Broker.

An example of a subscription is shown in Figure 127. The subscription refers to task entities and more specifically to receive updates upon changes on the "status" attribute.

[65] https://shop4cf.github.io/data-models/

```json
{
  "description": "Notify MPMS of tasks status changes",
  "type": "Subscription",
  "entities": [{"type": "Task"}],
   "watchedAttributes": [
      "status"
    ],
  "notification": {
            "format": "normalized",
            "endpoint": {
                "uri": "http://localhost:8080/tasks/status",
                "accept": "application/json"
            }
  },
   "@context": [
        "https://smartdatamodels.org/context.jsonld",
        "https://raw.githubusercontent.com/shop4cf/data-models/master/docs/shop4cfcontext.jsonld"
    ]
}
```

Figure 127: Example of a Context Broker subscription. Subscribes on changes on the "status" attribute of the "task" entity (according to datamodel specification).

Once subscriptions have been created, the main production processes can start, in which messages between MPMS and the Context Broker are exchanged. Figure 128 illustrates an example. When a task requires to be delivered to an agent through the Context Broker, a task entity is posted upon task instance creation. When the actual work has finished, the task status attribute of the task entity is updated (e.g., "completed") by the external component which controls the agent. Per subscription, the application receives a notification with the update and completes the task through process engine's API (or any specification).

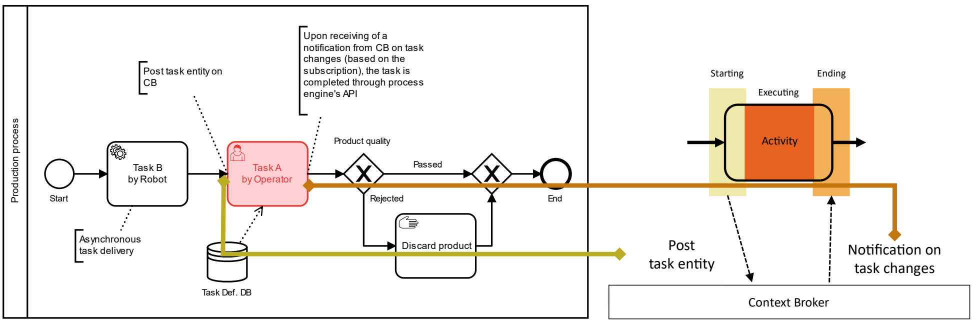


Figure 128: Example of production process in which task delivery is performed through Context Broker.

An example of a task entity, posted upon task instance creation, is shown in Figure 129. Note that its initial task status is set to "pending" (per datamodel specification). When the value changes, notifications are sent to the modules which have subscribed to these changes.

```
{
    "id": "urn:ngsi-ld:Task:bos:e0c4-6f0e",
    "type": "Task",
    "isDefinedBy": {
        "type": "Relationship",
        "object": "urn:ngsi-ld:TaskDefinition:bos:TABO"
    },
    "involves": {
        "type": "Property",
        "value": [
            {
                "type": "Relationship",
                "object": "urn:ngsi-ld:Resource:bos:operator_x"
            }
        ]
    }
   "happensAt": {
        "type": "Property",
        "value": [
            {
                "type": "Relationship",
                "object": "urn:ngsi-ld:Location:bos:workcell_1",
                "locationFunction": {
                    "type": "Property",
                    "value": "source"
                }
            },]
    },
  "workParameters": {
        "type": "Property",
        "value": {"batch_no": "34"}
    },
    "status": {
        "type": "Property",
        "value": "pending",
        "observedAt": "2020-12-01T11:23:19Z"
    },
    "outputParameters": {
        "type": "Property",
        "value": {
            "percentageCompleted": 0
        },
        "observedAt": "2020-12-01T11:23:19Z"
    },
    "@context": [
        "https://smartdatamodels.org/context.jsonld",
        "https://raw.githubusercontent.com/shop4cf/data-
                models/master/docs/shop4cfcontext.jsonld"
    ]
}
```

Figure 129: Example of a task entity to be posted on Context Broker.

Note that other type of messages can be exchanged as well, e.g., alerts, not only task-related information.

6.2.4 Architecture model of advanced MPMS

Having discussed the operationalization of the individual conceptual design artefacts, the functionality and the specification of the integrated solution is completed. Figure 130 provides the architecture model of an advanced manufacturing process management system, the elements of which are described in Table 24.

Table 24: Description of advanced MPMS elements.

Advanced MPMS elements			**Description**
Definition Tools	Process definition	Recipe controller	Covers functionality to specify manufacturing processes in terms of process models, i.e., what is the sequence of activities, which resources are involved (specifications of roles) and where they take place. It also includes the recipe controller model and the exception handling constructs.
		Exception handling constructs	
	Task definition		Covers functionality to specify tasks to be executed by (a team of) agents. Specification includes task input/output parameters, required roles, etc.
	Resource Definition		Covers functionality to specify the characteristics of resources involved in manufacturing processes, with a focus on agents that perform activities. Specification includes skills, authorization, etc.
	Location definition		Covers functionality to specify the entire production area/site in terms of physical layout.
	Definition data	Process/Task Def. data	Stores specifications of processes and tasks. Can be physical databases or file repositories. For instance, process definitions can have the form of BPMN/XML files.
		Resource Def. data	Stores specification of resources.
		Location Def. data	Stores specification of resources.
Process Enactment Service	Process Engine		Responsible to automatically control the actual execution of process definitions.
	Core application	Agent allocation	Covers functionality for selecting eligible (team of) agent(s) to perform a task.
		Task delivery	Covers functionality to deliver tasks to agents (both human and auto agents).
		Recipe system	Covers functionality to provide synchronization mechanism of processes.
		Exception handling	Covers functionality to handle exceptions.
	(Execution) Data	Engine data	Persists process engine data (e.g., running process instances, process variables, etc.)
		Application data	Stores application data (e.g., recipe's system persisted data, etc.).
Auxiliary tools	Administration Tools		Covers functionality to manage applications users (e.g., groups, authorizations, etc.)
	Monitoring Tools		Provides process monitoring and covers functionality to manage processes.
Interfaces	IF1		Provides interfacing between Definition Tools and Process Enactment Service. For instance, it can take the form of a file repository where process models are stored to be accessed by the Process Engine.
	IF2		Provides interfacing between Core Application and agents (both human and auto agents) to deliver tasks (through middleware technologies). Technically, it is to the systems that control those agents (e.g., a robot controller).
	IF3		Provides interfacing between Core Application and other enterprise information systems/other systems (through middleware technologies).
	IF5		Provides interfacing between the process enactment service and the Auxiliary Tools.

A few design decision/explanation points with respect to the architecture model of Figure 130:

Design decision 1): As the architecture model of advanced MPMS is based on the MPMS specification (Chapter 5), it keeps the same structure of Figure 103. That also means that its design is based on the three references architectures introduced in Section 5.3.2.

Design decision 2): Following the same reasoning as Design decision 1, interfaces IF1/2/3/5 are highlighted and keep the labeling of the WfRM.

Design decision 3): Data stores are included (compared to Figure 103), according to the detailed models of the Mercurius reference architecture (Grefen & Remmerts De Vries, 1998).

Design decision 4): The Recipe controller process model (Figure 117) and the exception handling modeling constructs (Figure 118 and Figure 119) are included in the Process Definition tool as should be used during process modeling.

Design decision 5): Process and Task definition data are conceptually placed under one data store, as these two concepts are closely related (i.e., a process is a series of tasks). Technically, these can be different stores.

Design decision 6): Def. data as specification of IF1 is not a separate data store, but it aggregates the three data stores of Definition Tools. It is drawn there to highlight that it can play the role of the IF1 for interfacing Definition Tools and Process Enactment Service.

Design decision 7): The Service/Integration layer is Figure 103 is further elaborated to specific connection points as different integration methods between MPMS modules and external systems are required.

Design decision 8): Integration to data stores for retrieving product related or order related information is omitted for simplicity reasons. The integration to such data stores can be achieved with the data connectors of the process engine or the core application.

Design decision 9): Internal interfacing between Process Engine and Core application is not specified with specific points, as there can be various ways to achieve this (e.g., through process engine's API or other connectors).

Design decision 10): IF5 does not have specific integration points with Process Enactment Service, as Auxiliary tools (i.e., Administration and Monitoring Tools) may connect with Process Engine (e.g., to show process instances) and/or Core application (e.g., to show application data). Typically, IF5 connects to the data store of process engine (and not directly to process engine), but to keep the interfacing flexible, it is not specified further.

Design decision 11): The Recipe system module is considered as an extension module of the workflow enactment server architecture of Mercurius (see Fig.7 of (Grefen & Remmerts De Vries, 1998)), similarly to the Exception handling module as discussed in Section 5.3.3.4.

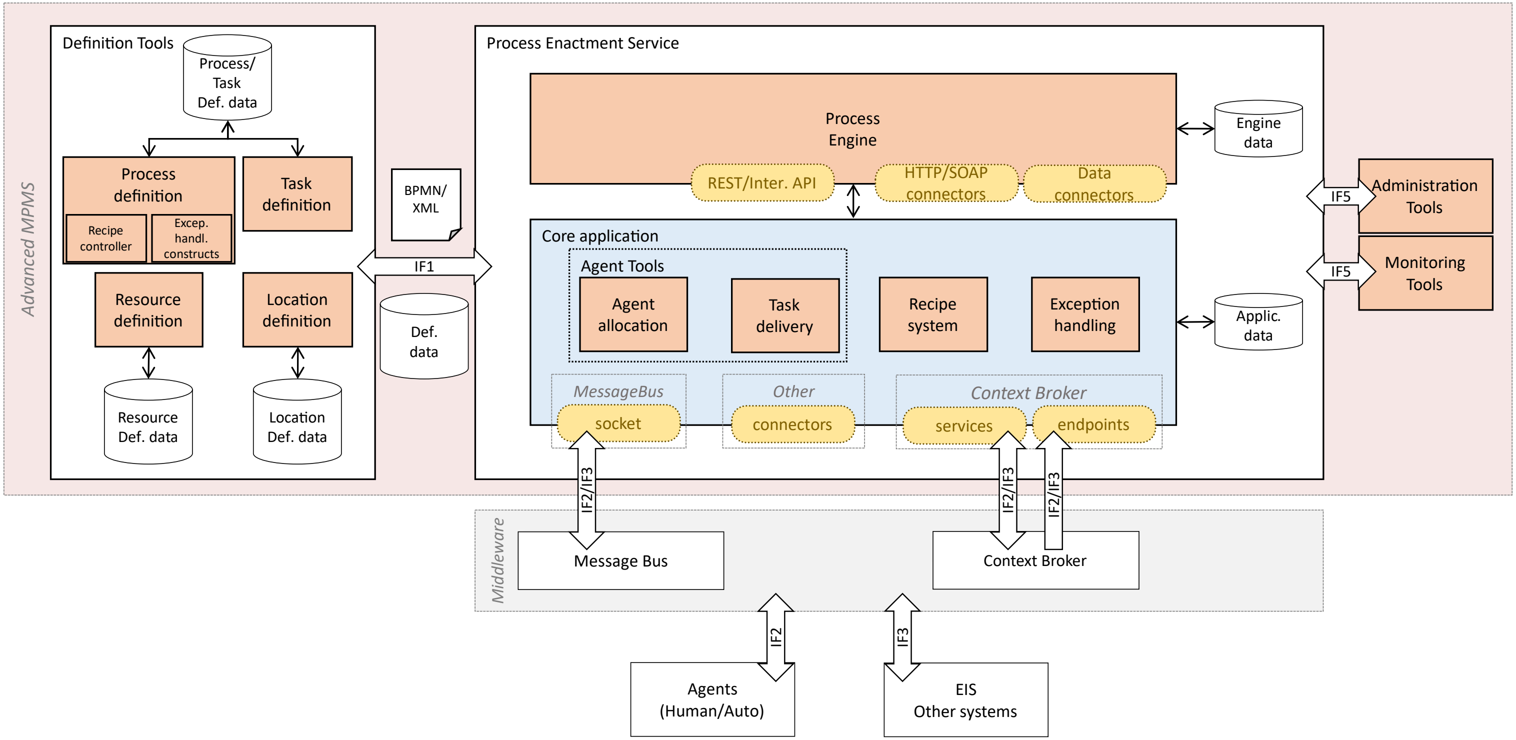


Figure 130: Architecture model of advanced MPMS.

6.3 Realization

The previous section presented the architecture of an advanced Manufacturing Process Management System (MPMS) that applies BPM theories in response to complexity issues in smart manufacturing. To realize such a system, an existing BPMS is chosen as a basis, presented in Section 6.3.1. On top of this basis, extra components shall be realized to provide the advanced functionality that this research proposes (as discussed in Section 6.2.4). The complete realized system is presented in Section 6.3.2. Finally, a deployment diagram of the system is presented in Section 6.3.3, as part of the physical view of K4+1 framework (see Figure 81).

6.3.1 Existing BPM tooling

The selection of an existing BPM system was driven by a major requirement of the HORSE project for providing open-source solutions. Various open-source systems exist with different technologies, support and extensibility options. A decision was made upon the beginning of the HORSE project (2015) to explore Camunda Platform[66] (Camunda BPM back then), among other options such as Activiti[67], Flowable[68], Bizagi[69].

Camunda Platform, from the Camunda[70] organization, is a leading workflow and decision automation platform for end-to-end business process orchestration. It is widely used in financial and insurance services, media and entertainment, technology and telecommunications sectors, by clients such as Deloitte, Allianz, Warner Music, Deutsche Telekom, etc. In a broad sense, it covers the main phases of the Business Process Management (BPM) cycle, i.e., process design, automation, monitoring and improvement. It offers native support for the latest BPMN, DMN and CMMN standards. The Community Edition, licensed under the Apache License, provides a highly extensive and scalable platform based on open-source components and a developer-friendly approach, with detailed documentation and a vibrant community. While it does not include all features of the Enterprise Edition that the organization offers (for obvious commercial reasons), it is a powerful and popular tool among companies and researchers to use and extend a workflow management/BPM system.

6.3.1.1 Camunda architecture

Camunda Platform features components for process modeling, execution and monitoring (with optimization component offered in the Enterprise edition). Figure 131 illustrates the overview of the main components, which are described below:

- **Modeler**: Modeling tool for BPMN 2.0 and CMMN 1.1 diagrams as well as DMN 1.3 decision tables. It comes as a user-friendly desktop application[71] licensed under the bpmn.io license (bpmn.io[72] is an open-source project for the modeling

[66] https://camunda.com/platform-7/
[67] https://www.activiti.org/
[68] https://www.flowable.com/
[69] https://www.bizagi.com/en
[70] https://camunda.com/
[71] https://docs.camunda.org/manual/7.15/modeler/
[72] https://bpmn.io/

framework and toolkits). A cloud-based solution, called Cawemo[73], is also available to ease the collaboration and business-IT alignment.

- **Process Engine**: The process engine is a Java library responsible for executing BPMN 2.0 processes, CMMN 1.1 cases and DMN 1.3 decisions. It has a lightweight Plain Old Java Object (POJO) core and uses a relational database for persistence. Object–relational mapping (ORM) mapping is provided by the MyBatis mapping framework.
- **REST API**: It allows using the process engine from a remote application or a JavaScript application.
- **Tasklist**[74]: A web application for human workflow management and user tasks that allows process participants to inspect their workflow tasks and navigate to task forms in order to work on the tasks and provide data input.
- **Cockpit**[75]: A web application for process monitoring and operations that allows for searching for process instances, inspecting their state and repairing broken instances.
- **Admin**[76]: A web application that allows for managing users, groups and applications authorizations.

[73] https://camunda.com/platform-7/cawemo/
[74] https://docs.camunda.org/manual/7.15/webapps/tasklist/
[75] https://docs.camunda.org/manual/7.15/webapps/cockpit/
[76] https://docs.camunda.org/manual/7.15/webapps/admin/

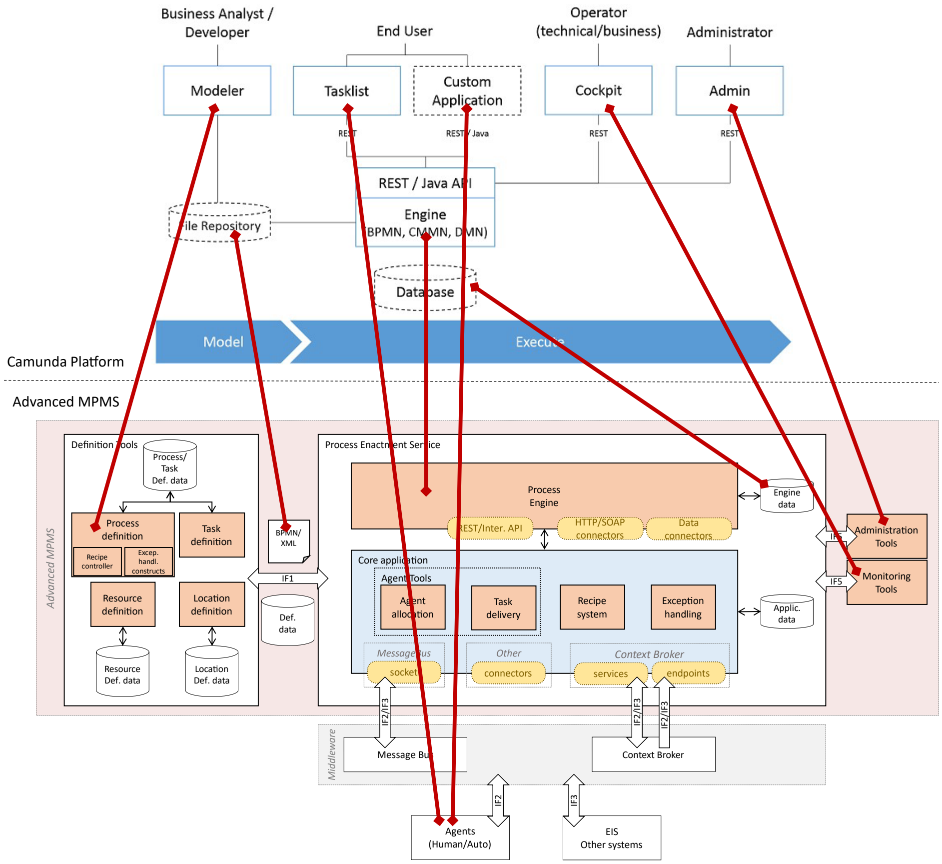


Figure 131: Camunda Platform architecture[77] and mapping to advanced MPMS architecture.

Figure 131 also illustrates the mapping of the Camunda Platform architecture components onto the designed advanced MPMS architecture (Figure 130). As can be seen, three modules of the Definition Tools and the Core application with its interfaces from the MPMS architecture are not covered by the Camunda Platform architecture. These modules shall be realized with different technologies, as discussed in Section 6.3.2.

6.3.1.2 Camunda process engine

The process engine of Camunda Platform is a core component that provides the automation and is the most complex one. Thus, it is further elaborated in this Section.

Camunda Platform is a Java-based framework. The main components are written in Java and there is a general focus on providing Java developers with the tools they need for designing, implementing and running business processes and workflows on a Java virtual machine (JVM). Nevertheless, to make the process engine technology available to non-Java developers, the platform also provides a REST API which allows developers to build applications connecting to a remote process engine.

[77] https://docs.camunda.org/manual/7.15/introduction/

The process engine architecture is shown in Figure 132, with its main components explained below:

- **Process Engine Public API**: Service-oriented API allowing Java applications to interact with the process engine. The different responsibilities of the process engine (i.e., Process Repository, Runtime Process Interaction, Task Management, etc.) are separated into individual services. The public API features a command-style access pattern: Threads entering the process engine are routed through a Command Interceptor which is used for setting up Thread Context such as Transactions.
- **BPMN 2.0 Core Engine**: This is the core of the process engine. It features a lightweight execution engine for graph structures (PVM - Process Virtual Machine), a BPMN 2.0 parser which transforms BPMN 2.0 XML files into Java Objects and a set of BPMN Behavior implementations (providing the implementation for BPMN 2.0 constructs such as Gateways or Service Tasks).
- **Job Executor**: The Job Executor is responsible for processing asynchronous background work such as Timers or asynchronous continuations in a process.
- **The Persistence Layer**: The process engine features a persistence layer responsible for persisting process instance state to a relational database. The MyBatis mapping engine is used for object relational mapping.

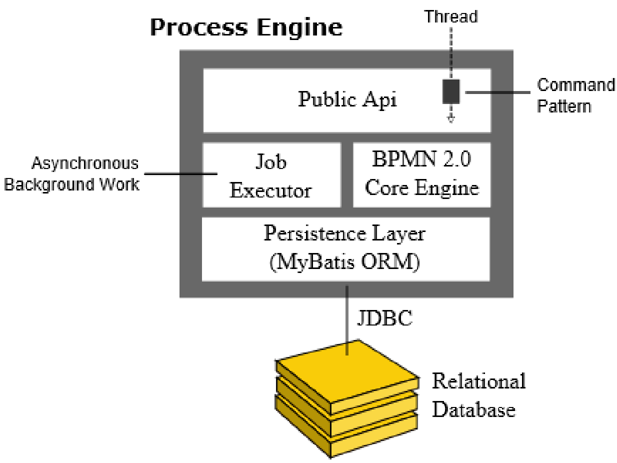


Figure 132: Camunda process engine architecture[78].

6.3.2 Advanced MPMS components

Camunda platform, as the basis of the advanced MPMS, covers core functionality but not the entire functionality to support process management in smart manufacturing (as illustrated in Figure 131). To realize a system based on the architecture of Figure 130, extensions on existing components and extra components are built. Figure 133 illustrates the overview of the developed components of an advanced MPMS, which are described below:

[78] https://docs.camunda.org/manual/7.15/introduction/architecture/

- **Definition Tool**: Captures requirements of manufacturing scenarios and translates them into definitions in accordance with the data models (Section 6.2.3.1). It is implemented as MS Access[79] tool. It provides the functionality to define process requirements, tasks, resources and locations. These definitions are also stored in a PostgreSQL[80] database, to be accessed and used by the execution (runtime) components.
- **Modeler**: The default Camunda Modeler to draw BPMN, CMMN models and DMN diagrams and tables. As the Modeler traditionally is used by business roles (e.g., analysts, process modelers), user-friendliness and collaboration are important aspects. Thus, available layout plugins are used. For instance, a layout plugin that shows tooltips[81] with basic details of an element, without having to open the properties panel. Extensions for embedding comments on elements[82] is available, as well. Element templates[83] are also available to provide extra configurations, such as properties on a Service Task to automatically send an email or connection details of an external (web) service. Note that the Recipe Controller process model and the exception handling constructs are readily available as BPMN process models to be altered per need and deployed in application projects.
- **Process Engine**: The default Camunda process engine to enact the modeled processes. It includes extensions to provide delegate code (listeners) upon starting and ending phases of tasks (see Section 6.2.1.1). Process engine data are persisted in a PostgreSQL database.
- **Core application**: A Spring Boot[84] application written in Java, as a standalone application that can run easily. It handles the business logic of the process models and includes functionality for:
    - **Agent allocation**: Implements the mechanism for selecting eligible (team of) agent(s) to perform a task (Erasmus, 2019), as modelled in Figure 55.
    - **Task delivery**: Implements the handling of task assignment to agents (as discussed in Section 6.2.1.1 and Section 6.2.3.2).
    - **Recipe system**: Implements the synchronization mechanism of processes (as discussed in Section 6.2.1.2).
    - **Exception handling**: Implements the business logic of exception handling (as discussed in Section 6.2.2).
    - **Interfaces**: Connection points to middleware technologies (as discussed in Section 6.2.3.2) and other systems.
- **Tasklist UI**: Application(s) to present tasks to human agents. Available as a web application (default Camunda Tasklist) and as an Android smartwatch application.
- **Cockpit**: The out-of-the box Camunda Cockpit web application for process monitoring. A plugin to visualize process variables onto the process model instances has been created as well (see Section 7.1 for examples).
- **Users Admin**: The out-of-the box Camunda Admin web application for managing applications users.

---

[79] https://www.microsoft.com/en-ww/microsoft-365/access
[80] https://www.postgresql.org/
[81] https://github.com/viadee/camunda-modeler-tooltip-plugin
[82] https://github.com/camunda/camunda-modeler-plugins/tree/master/bpmn-js-plugin-embedded-comments
[83] https://github.com/camunda/camunda-modeler/tree/master/docs/element-templates
[84] https://spring.io/projects/spring-boot

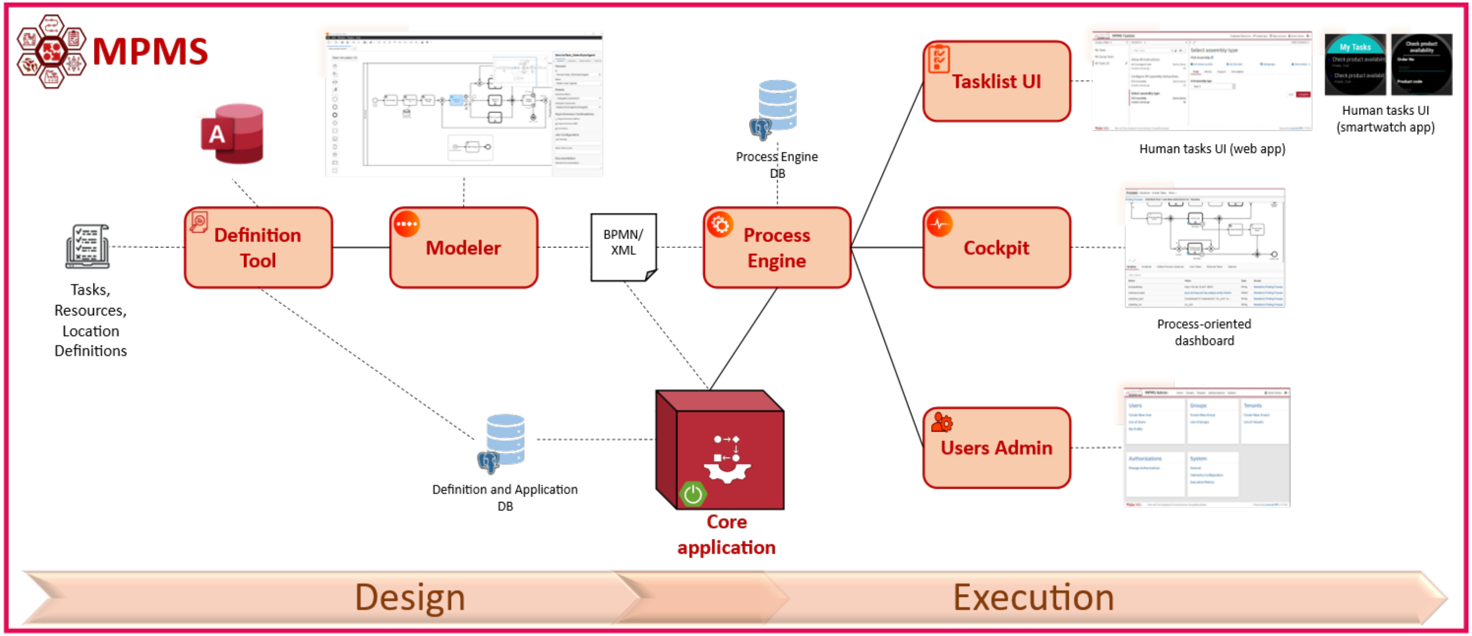


Figure 133: Overview of the developed components of the realized advanced MPMS.

Examples of illustrating the use of the components are provided in Section 7.1 through the demonstration of the system in real-world settings of pilots.

6.3.3 Deployment diagram

Camunda Platform is a flexible framework which can be deployed in different scenarios. It can be used both as a standalone process engine server or embedded inside custom Java applications. The embeddability requirement is at the heart of many architectural decisions within Camunda Platform. The most common deployment scenarios are illustrated in Figure 134.

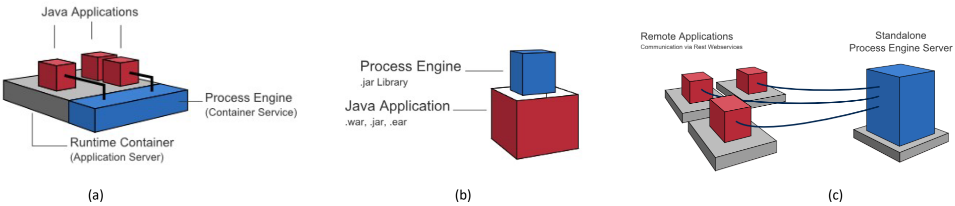


Figure 134: Camunda Platform deployment scenarios[85]: (a) Shared, Container-Managed process engine, (b) Embedded process engine, (c) Standalone (remote) process engine server

In the first deployment scenario (Figure 134(a)) the process engine is started inside the runtime container (Servlet Container, Application Server, etc.). The process engine is provided as a container service and can be shared by all applications deployed inside the container. The concept can be compared to a Java Message Service (JMS) Message Queue which is provided by the runtime and can be used by all applications. There is a one-to-one mapping between process deployments and applications: the process engine keeps track of the process definitions deployed by an application and delegates execution to the application in question. In the second deployment scenario (Figure 134(b)) the process engine is added as an application library to a custom application. This way, the process engine can easily be started and stopped with the application lifecycle. It is possible to run multiple embedded process engines on top of a shared database. In the third deployment scenario (Figure 134(c)) the process engine is provided as a network service. Different applications running on the network can interact with the process engine through a remote communication channel. The easiest way to make the process engine accessible remotely is to use the built-in REST API. Different communication channels such as SOAP Webservices or JMS are possible but need to be implemented by users.

The embedded process engine option has been selected, due to its flexibility of implementing the core application of MPMS as a Spring Boot application. To ease the deployment and control of the software, the software packages are provided based on the containerization approach, as has been discussed in Section 5.4.3.2 (see Figure 111). Using Docker[86] as the containerization technology, MPMS (runtime) packages are deployed as shown in Figure 135.

[85] https://docs.camunda.org/manual/7.15/introduction/architecture/
[86] https://www.docker.com/

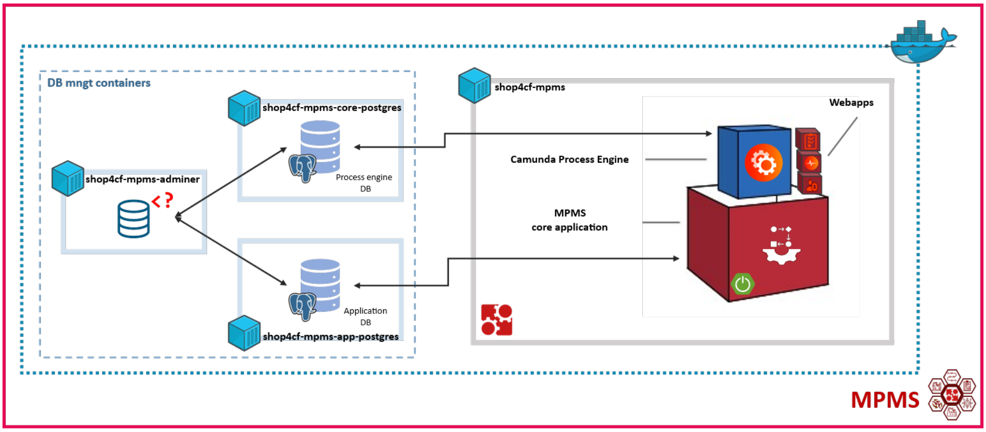


Figure 135: Deployment diagram of advanced MPMS as Docker containers.

More discussion on deployment (physical view of K4+1) is provided in Section 7.1, through the demonstration of system in real-world settings of pilots, as part of entire CPS.

6.4 Chapter conclusion

The identified issue of complexity in manufacturing operations, introduced in Chapter 1, has been approached from three different perspectives (discussed in Section 2.1.6). Conceptual design artefacts are developed to tackle each of these perspectives, as presented in Chapters 3, 4 and 5. However, for addressing the general issue, an integrated solution is required.

Thus, this chapter presented a response to RQ4 on how an advanced manufacturing process management system can support the complexity tackling in smart manufacturing. An architecture model of a system has been designed, as the ensemble of the operationalized artefacts of the individual conceptual designs. The design has been inspired by existing WfMS and BPMS reference architectures, covering though missing functionality for application in smart manufacturing. Thus, it acts as a blueprint on how to realize a system to support modeling of complex production scenarios, to support operational exception handling, and to enable horizontal and vertical integration of manufacturing operations.

A possible realization of the architecture model is presented as well, to prove that the designed artefact is viable. The realized system has been demonstrated in various pilots within the three projects. Demonstrations at three of these pilots (analyzed in Section 2.2) are presented in Section 7.1. The evaluation of the system is discussed in Section 7.2.

CHAPTER 7

Demonstration and evaluation

Following the DSRM (Figure 11), the developed artefacts shall be demonstrated and evaluated. Thus, this chapter discusses these two phases. Section 7.1 presents illustrative examples of the showcased implemented solutions. The advanced MPMS developments have been applied as part of complete CPS solutions. Section 7.2 discusses the evaluation of the designed solutions. Section 7.3 concludes the chapter.

7.1 Demonstration

The three EU projects (introduced in Section 2.2.1) provided plenty of opportunities to apply and demonstrate the designed solutions. The three pilot cases discussed in Section 2.2, as part of the problem analysis, are used here as testbeds to demonstrate the realized artefact(s) in real-world operational environments. Thus, the next three subsections present the demonstrated solutions. An overview of the application of the implemented MPMS in 14 pilots in total is provided in Appendix K.

The presentation of each pilot follows the same structure:

- Scenario(s) description, with a focus on application of the MPM designed artefact(s). The analysis process models (from Section 2.2) have been transformed into executable ones based on the approach documented in Vanderfeesten et al. (2019); The executable models incorporate the realized artefacts (presented in Section 6.2);
- Details on the developed CPS, in which MPMS is embedded, as integrated solution;
- Illustration of the physical demonstrators to showcase the results. A representative set of results is selected (for reasons of brevity). More information for published demonstrator media is available in Appendix J.

7.1.1 TRI

TRI, as a manufacturer with the ambition to transition into smart manufacturing[87], provided various options to test new technologies and approaches. All three intervention scenarios (highlighted in Figure 30 and described in Section 2.2.2.5) were tested at great extent, in the context of the HORSE project, providing useful results.

7.1.1.1 Scenario(s) description

The three intervention scenarios are described individually in the following three subsections.

[87] https://youtu.be/JBodoko84jc - TRI IROS

7.1.1.1.1 Single tool assembly

The tool assembly process (Section 2.2.2.5.1) has undergone significant changes compared to the as-is situation. An AR system projects assembly instructions on a worktable to assist inexperienced operators. In parallel, a mobile robot fetches the necessary tools from the storage. MPMS orchestrates the activities, as modeled in Figure 136. More specifically, task messages are delivered to the AR system through a message bus middleware. Similarly, task messages are delivered to the controller of the mobile robot. Errors during the AR task are propagated to a teamleader who has the experience to either fix the errors or cancel the process.

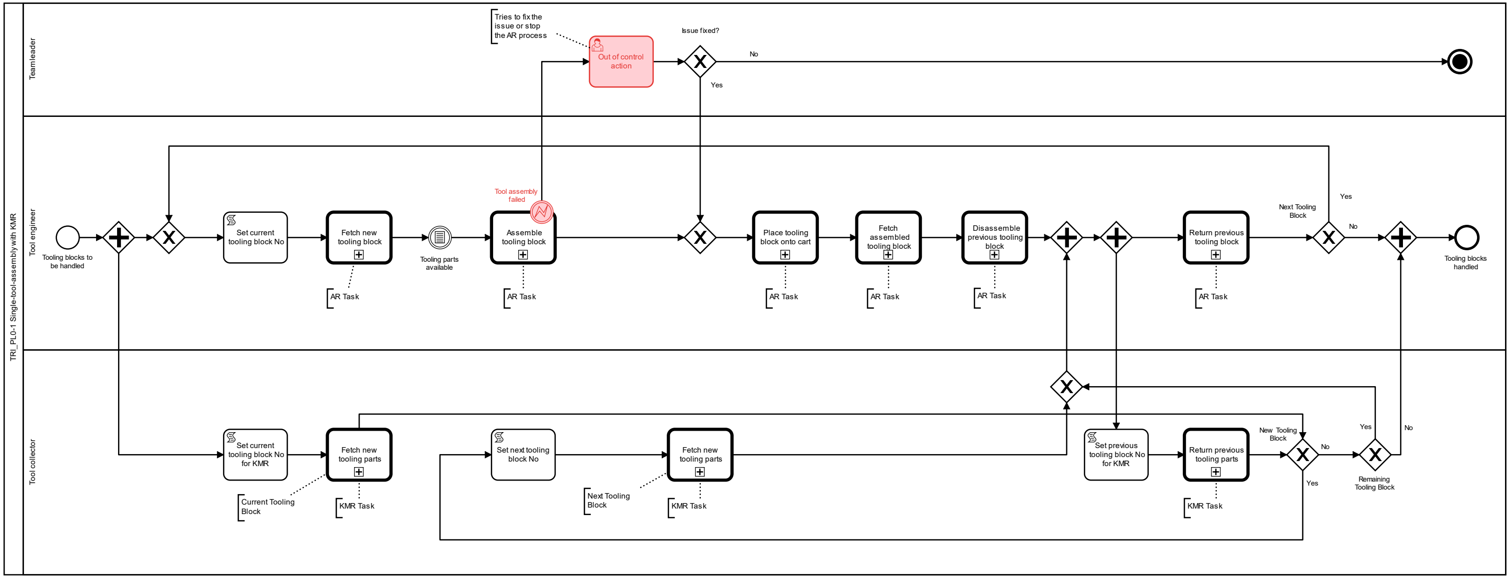


Figure 136: Tool assembly process at TRI with clear indication of parallel activities for AR support for assembly instructions (middle swimlane) and mobile robot for tool parts collection (bottom swimlane), modelled and orchestrated by MPMS. An example of exception handling is shown as well (highlighted in red).

7.1.1.1.2 Profile stacking

The structured and layered stacking of profiles into bins (Section 2.2.2.5.2) was also automated with the use of a conveyor belt and a robot arm. The robot arms picks-up a number of profiles and places them, layer-by-layer, into a bin. In between layers, a separator has to be placed, so the profiles are easily picked-up again from the next robot in P2 area. The process has been modeled in MPMS, as shown in Figure 137. The iteration of tasks by the robot arm is handled by MPMS (purple-highlighted sequential multi-instance subprocess, according to the pattern of Figure 59). A mobile platform is responsible for the transportation of bins (both empty and full).

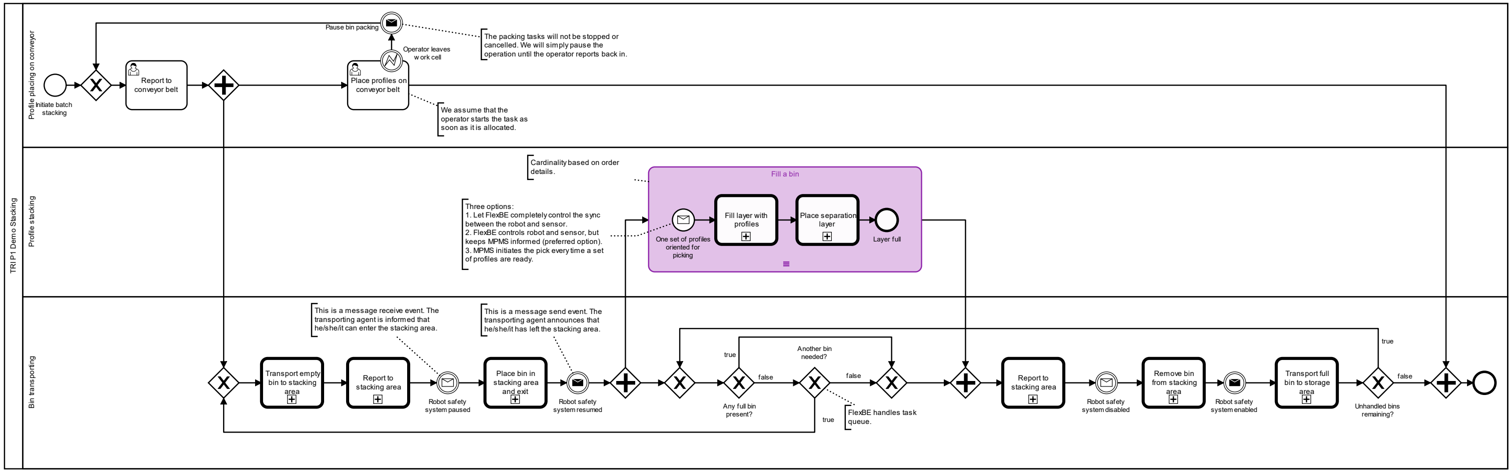


Figure 137: Profile stacking process at TRI by a robot arm. The highlighted subprocess (purple) shows the handling of multiple tasks by MPMS.

7.1.1.1.3 Loading and unloading of profiles

A robot arm with a special gripper is used to assist the loading and unloading of profiles taking place at P2 (Section 2.2.2.5.3). The orchestration of activities is managed by MPMS. The modeled loading process is shown in Figure 138. The green-highlighted task applies an agent allocation mechanism to select either a human operator or a robot to perform the hanging. The purple-highlighted sequential multi-instance task (calling the task delivery pattern of Figure 55) follows the iteration pattern of Figure 59. The unloading process is the counteractive part, omitted here for reasons of brevity.

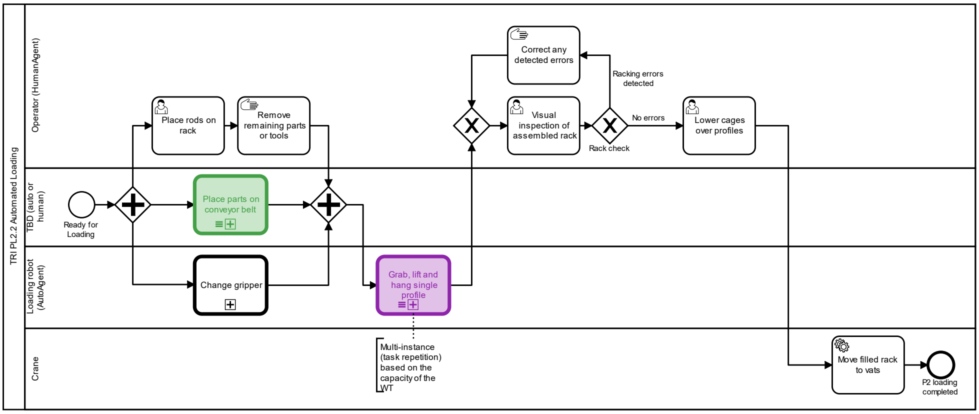


Figure 138: Loading of profiles onto racks process at TRI, performed either by human operator or robot (allocation mechanism highlighted in green). The purple highlighted task shows the multiple task handling by MPMS.

7.1.1.2 Developed CPS

MPMS has been applied at all TRI intervention scenarios to enact all the modeled processes presented in Section 7.1.1.1. As part of developed CPS, the system communicates with the new technologies according to the HORSE system architecture. Figure 139 shows the technology stack applied at all TRI scenarios, as a more concrete instantiation of the HORSE CPS technology stack shown in Figure 110.

The cyber-physical middleware is realized with a websocket message bus, based on OSGi. The role of the Hybrid Task Supervisor, responsible to design and synchronize execution steps by agents, is undertaken by FlexBe[88] software. It features OSGi plugins to existing OSGi nodes, ROS integration to robots and interfaces to industrial equipment. FlexBe was customized and extended by a project partner to be able to connect to other HORSE components. ROS (O'Kane, 2014) is a commonly used, open-source, meta-operating system for robots and provides functionality such as hardware abstraction, low-level device control, implementation of commonly used functionality, message-passing between processes, and package management. GPU Voxels (Hermann et al., 2014) has been used to provide advanced robot motion planning, covering safety functionality at local control level. To provide the processing power required by such software, a parallel computing platform and application programming interface model is required (nVidia CUDA (Storti & Yurtoglu, 2015)). Such a high-performance processing platform enables real-time 3D projections.

[88] http://wiki.ros.org/flexbe

KUKA Sunrise robotics interface has been used to provide all functions to operate the lightweight KUKA robots. Finally, a Light Guide System (LGS) was adopted and customized by a project partner to provide the AR support.

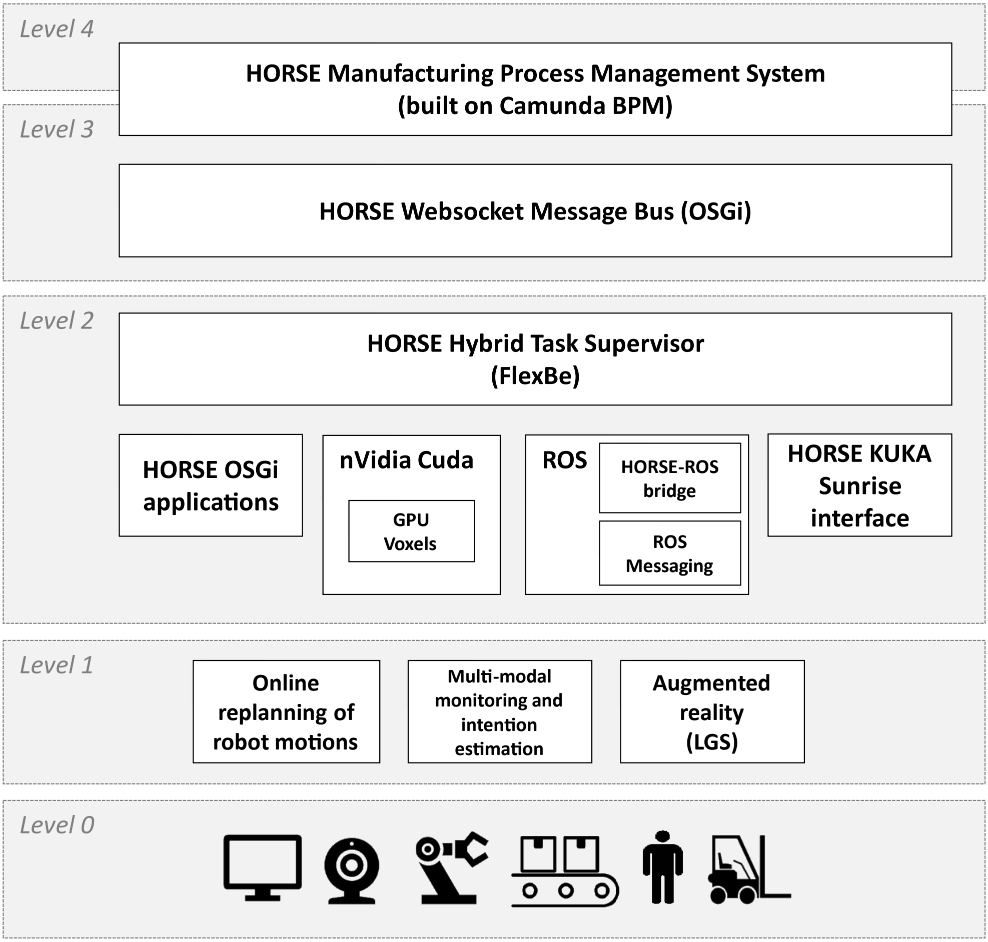


Figure 139: Technology stack of deployed CPS at TRI.

All the components have been deployed according to the topology of Figure 107. Figure 140 shows the deployment diagram of the CPS at TRI. For simplicity reasons, only the local deployment of components at the single tool assembly intervention scenario is shown. As can be seen, MPMS is deployed at a global domain level, able to orchestrate activities of all three local deployment solutions (consisting of heterogeneous technologies).

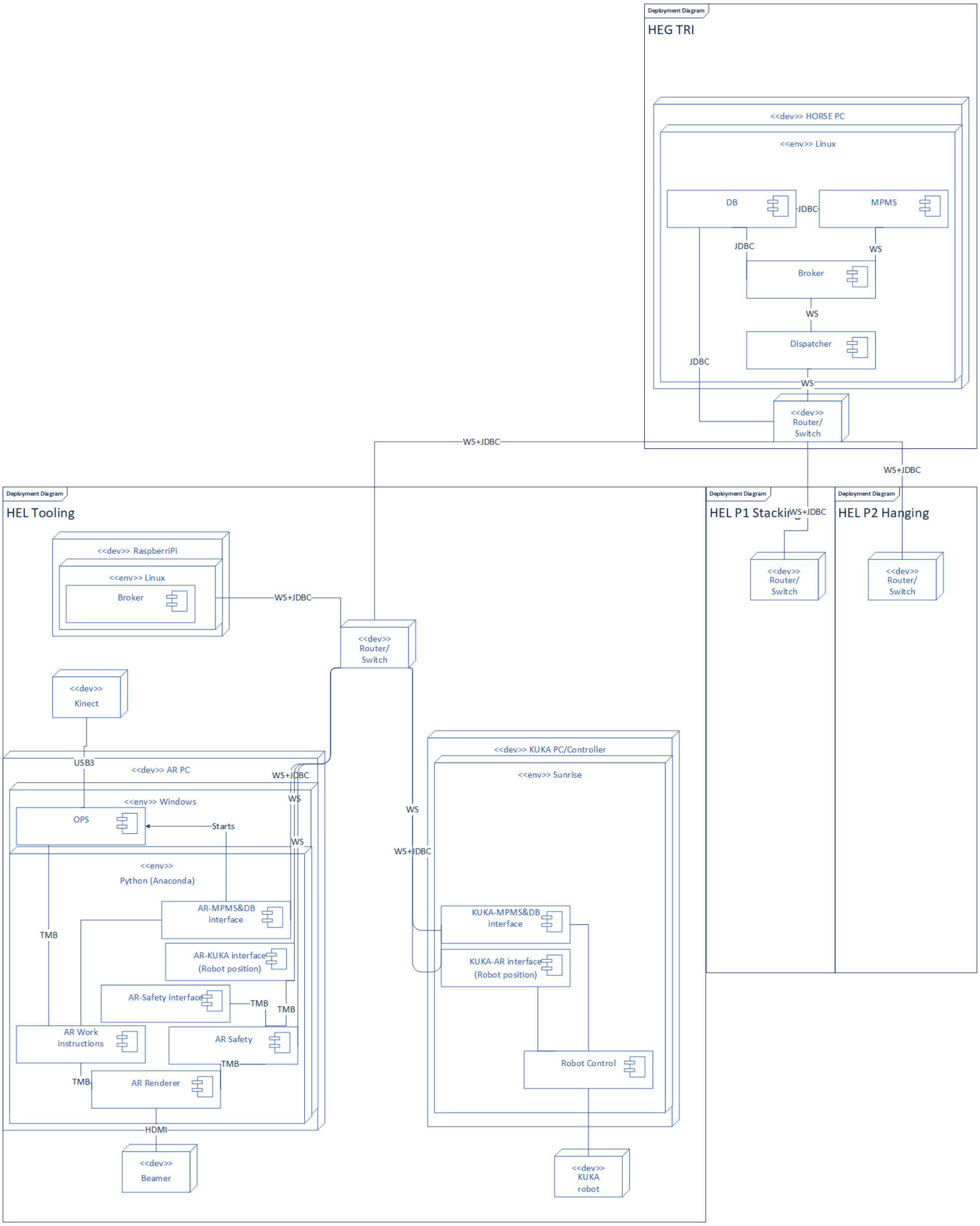


Figure 140: Deployment diagram of CPS at TRI, with one global domain and three local domains (European Dynamics, 2018).

7.1.1.3 Physical demonstrator

The following three subsections present a few details per intervention scenario of the physical demonstrators.

7.1.1.3.1 Single tool assembly

The tool assembly process starts with the operator selecting a production order through the MPMS Tasklist application, as shown in Figure 141. Each production order includes a set of tools to be assembled, presented to the operator.

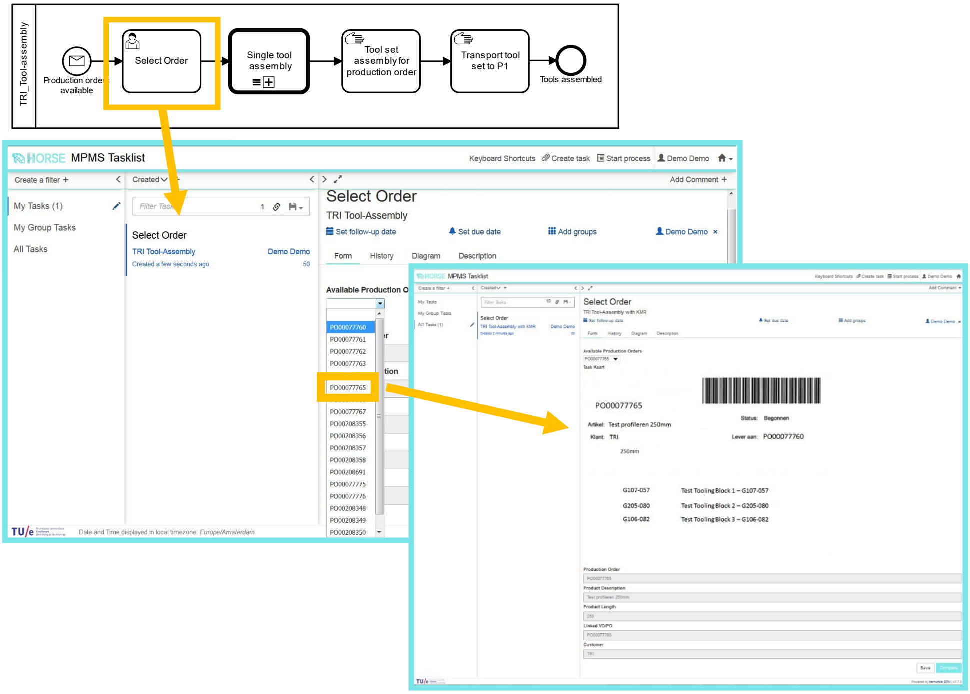


Figure 141: Tool assembly production order selection through MPMS Tasklist.

A snapshot of the actual single tool assembly assisted by AR is shown in Figure 142. As can be seen in the background, the mobile robot fetches the bins with tool parts (for the next assembly block) from the storage area, letting the operator to focus on the tool assembly process.

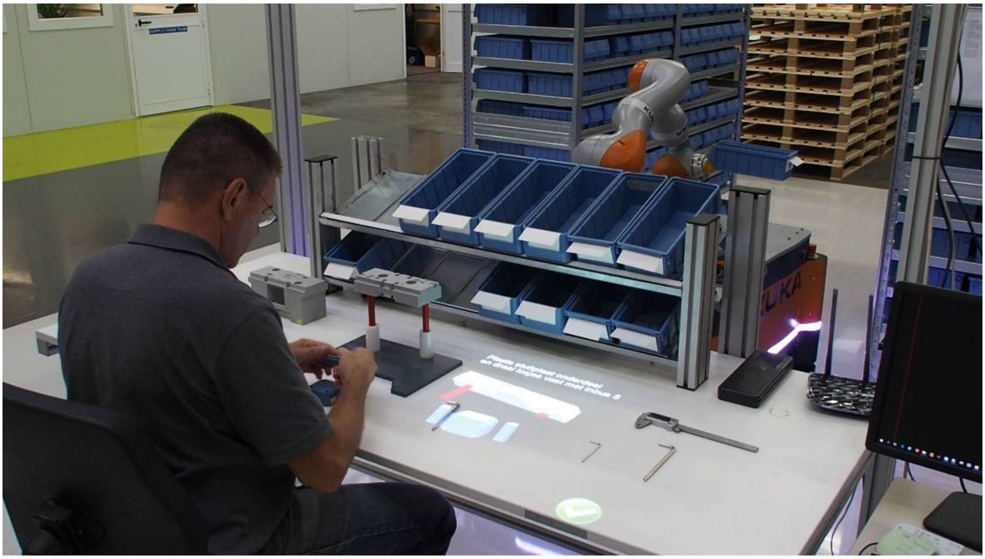


Figure 142: Single tool assembly process at TRI with AR support for assembly instructions and mobile robot for tool parts collection.

7.1.1.3.2 Profile stacking

Figure 143 illustrates the physical layout of the profile stacking demonstrator. The robot arm picks-up the profiles from the conveyor belt and places them into an empty bin, either being on the right or left side. Layer separators are placed by the robot arm as well. An AGV is responsible to transport the bins.

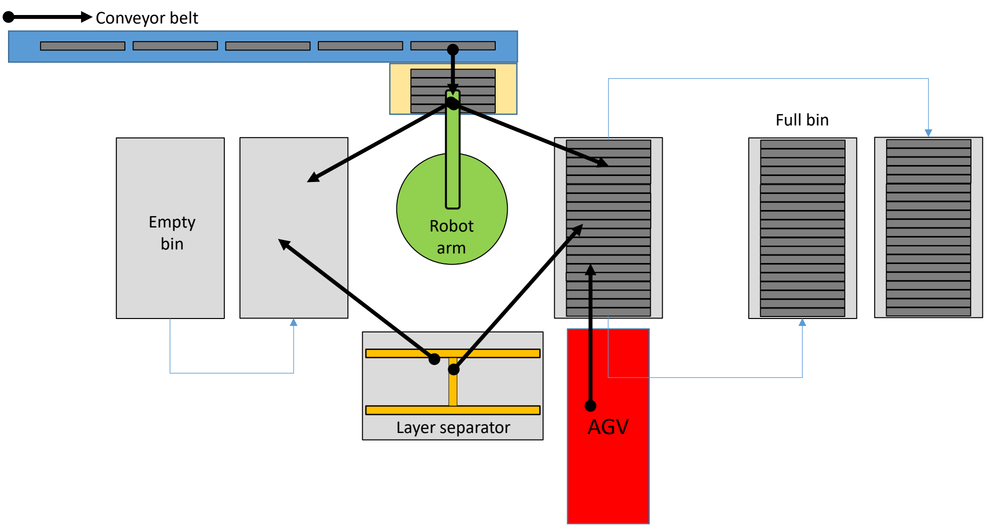


Figure 143: Physical layout of P1 profile stacking at TRI (source: HORSE project's material).

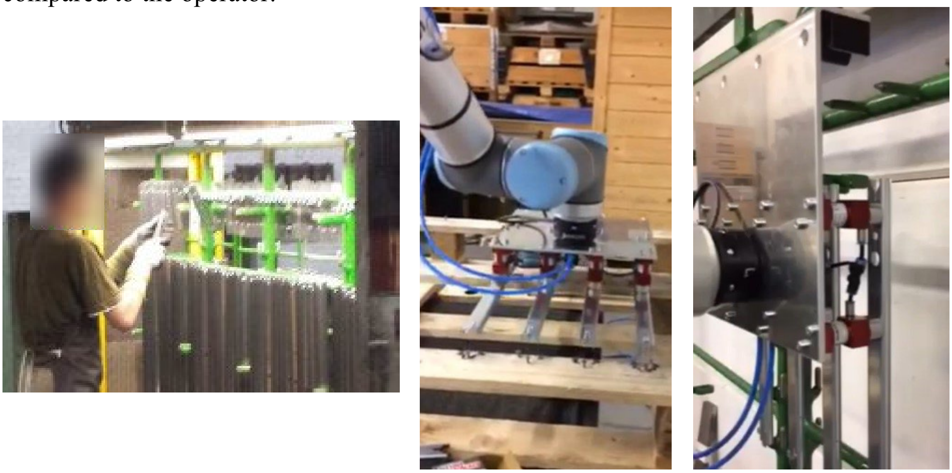
Figure 144(b) shows the stacking of profiles to the bin by the robot arm, in comparison to the unstructured placing in the current situation (Figure 144(a)). The use of the AGV was demonstrated in a different phase.

(a) (b)

Figure 144: Profile stacking into bins: (a) unorganized by human operators (current situation), (b) structured by robot arm (source: HORSE project's material).

7.1.1.3.3 Loading and unloading of profiles

Figure 145 showcases the hanging of profiles onto racks, either by the human (as performed in the current situation) or by the robot. It was proven that the robot was significantly slow compared to the operator.



(a) (b)

Figure 145: Profile hanging onto racks: (a) by human operator (as current situation), (b) by robot (source: HORSE project's material).

7.1.2 CPP

The demonstration at CPP took place in their Customer Experience Center in Venlo, the Netherlands, in the context of the EIT OEDIPUS project. The two intervention scenarios

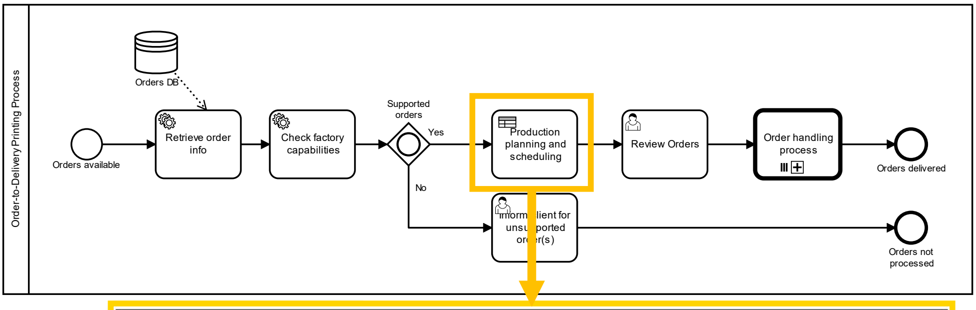
(Section 2.2.3.5) were demonstrated in two different phases, however, for the purposes of the current thesis they are treated as a single case study.

7.1.2.1 Scenario(s) description

The complete scenario refers to the production of books with the involvement of various resources:

- 1 cover printer “CP”, to print covers of books;
- 1 book-block printer “BBP” + 1 binder attached, the BBP prints bookblocks of books and with the loaded covers (produced by a CP) are automatically binded;
- 1 printer + 1 binder attached “PB”, to print any media that require binding;
- 1 collaborative robot arm mounted in front of the printer of the PB devices, to unload media from the high capacity stacker (HCS) of the PB onto designated trays;
- 1 AGV + a motorized robot arm “MRA” + 1 deposit tray mounted on it, to transport media (e.g., covers from CP to be loaded into BBP);
- 1 guillotine/trimmer, to trim media into the right size;
- 7 work-in-progress (WIP) trays, to deposit media to be picked-up by the robot arm or a human operator;
- 4 “virtual” printers, to demonstrate the routing of orders in different types of printers and the handling of multiple process instances;
- 1 human operator, equipped with a smartwatch to access tasks (e.g., loading covers into a BBP or taking corrective actions in case of exceptions).

A set of business orders are handled, consisting of many orderlines and print jobs. At orders level (see top process model of Figure 41), a production planning service is first called to determine the type of printers that should handle each orderline. The determination is based on a decision tree, modeled as a DMN decision table as shown in Figure 146. Note that the decision is taken on the type of the printer (e.g., cutsheet/black-white or continuous feed/full-color), and not on specific devices. The assignment to specific printers is performed with the task allocation mechanism just before a printing task has to be executed.



Determine Printer type for Orderline — Hit Policy: Unique

	When	And	Then	And	
	pages X copies (integer)	% FC pages (double)	printertype (string)	printercolortype (string)	Annotations
1	>=10000	-	"CF"	"FC"	
2	<10000	0	"CS"	"BW"	
3	<10000	(0..10]	"CS"	"FC-BW"	
4	<10000	>10	"CS"	"FC"	
+	-	-			

Figure 146: Decision table to determine the type of printer to handle an orderline, modeled in DMN.

Let us focus on the production of covers. The process model, shown in Figure 147, refers to a single print job, i.e., a single cover. The process highlights advanced MPMS developments such as agent allocation (green), synchronization points (blue) and exception handling (red).

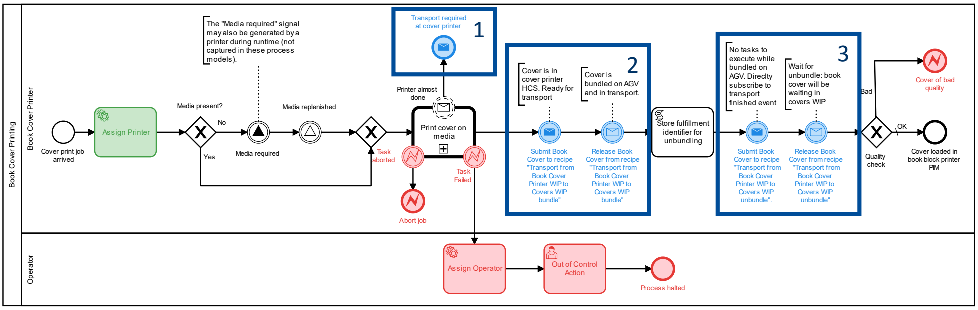


Figure 147: Book cover production process with highlighted advanced MPMS functionality: (green) agent allocation (green), (three) synchronization points (blue), exception handling (red).

Regarding the synchronization points, the process includes three main points. The first refers to raising a message when a cover is about to finish being printed (note that when many covers are being printed on the same printer, i.e., many task instances for the same printer, the recipe controller handles only one of them for avoiding redundant messages). This message triggers the AGV to go to the cover printer. The queue task management of the AGV is shown in Figure 148, in which corresponding synchronization points are highlighted as well.

The second synchronization point refers to bundling a group of covers onto the AGV (to be transported to a BBP). When each instance (cover) is ready, it informs the recipe controller (with a Submit message), which updates the state of the recipe accordingly. The corresponding recipe definition is listed in Table 25. According to it, when 5 covers (from 5 running instances) are printed, 1 AGV is ready to bundle them (with the use of the motorized arm). Note that the recipe system considers both cases, either the AGV is already present when the fifth cover is printed, or the AGV arrives at the printer at a later point. This is satisfied through the way the recipe controller continuously evaluates recipes (see Figure 117). When all covers are bundled on the AGV, the "waiting" state of each of their corresponding process instance is "released".

Table 25: Defined recipe for bundling covers onto the AGV (synchronization point 2 of Figure 147 and Figure 148).

Recipe name:	Transport from Book Cover Printer WIP to Covers WIP bundle					
Selector attribute:	None					
Input instance type	min	max	gen	relpol	mask	rel
Book Cover	1	5	●	LIFO	LAST	●
AGV	1	1	●	FIFO	FIRST	●

The third synchronization point is the counterpart of the second. More specifically, when the covers have been transported (and deposited) to the BBP, the recipe controller "releases" the continuation of their instances. This is defined by the recipe listed in Table 26.

Table 26: Defined recipe for unbundling covers from the AGV (synchronization point 2 of Figure 147 and Figure 148).

Recipe name:	Transport from Book Cover Printer WIP to Covers WIP unbundle					
Selector attribute:	Fulfillment ID of recipe Transport from Book Cover Printer WIP to Covers WIP bundle					
Input instance type	min	max	gen	relpol	mask	rel
Book Cover	1	5	○	FIFO	ALL	●
AGV	1	1	○	FIFO	ALL	○

Regarding the task management of the AGV (Figure 148), the process model follows the task queue management construct, as designed and presented in Section 3.3.3. Note that each task referring to the AGV or its mounted motorized arm is modeled as a subprocess, which calls the task delivery pattern (Section 3.3.1 and Section 6.2.1.1).

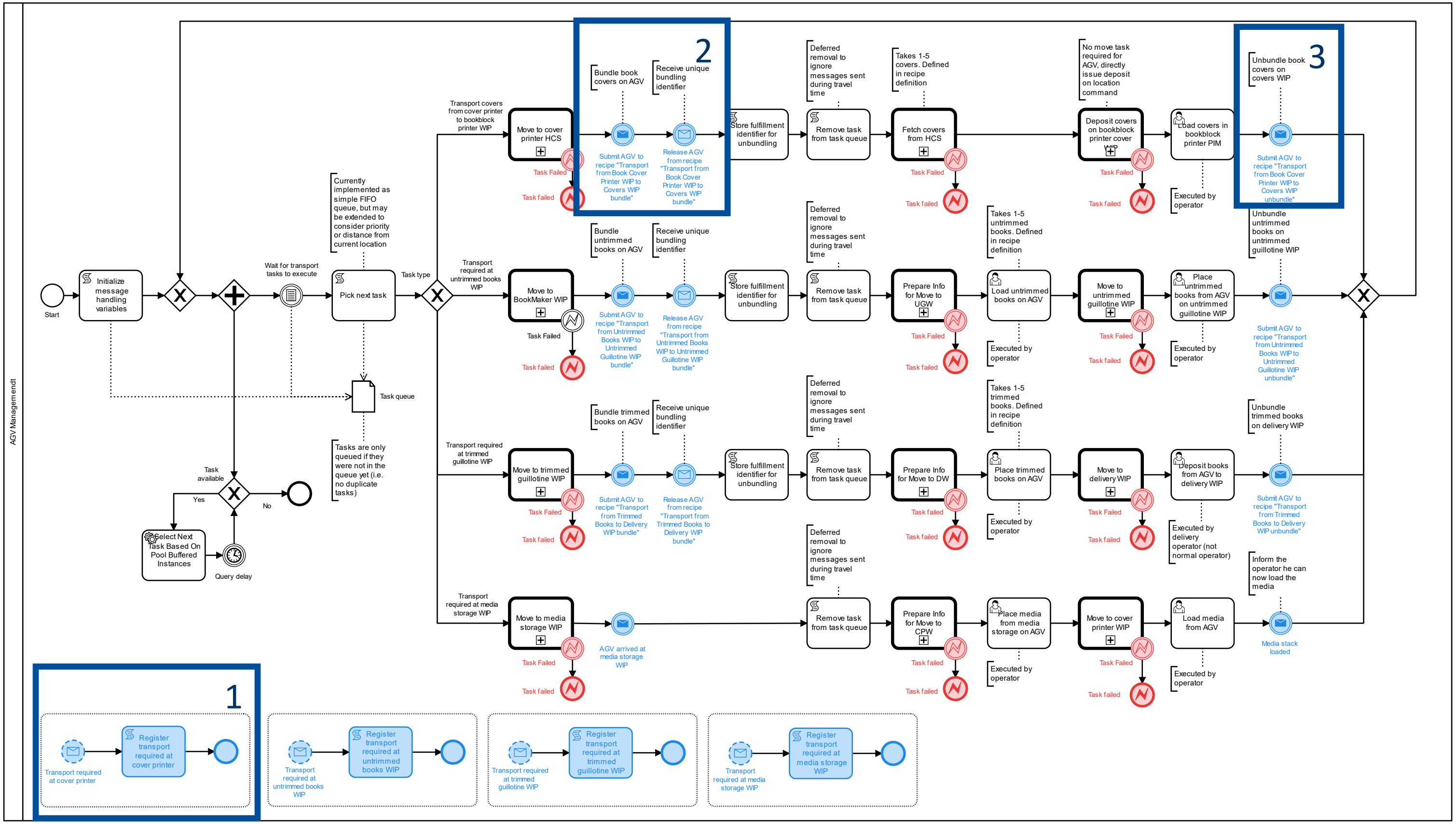


Figure 148: AGV (queue) task management with synchronization points (blue) and exception handling (red).

7.1.2.2 Developed CPS

A Printing Process Management System (PPMS), realized based on the architecture model of the advanced MPMS (Figure 130), provides global control of activities performed by various resources in different locations of the "print shop". This enables horizontal integration. The vertical integration, i.e., the actual control of resources, is achieved with the development of a cyber-physical system. The implemented CPS respects the global/local separation of concern (as discussed in Chapter 5). Its high-level architecture model is shown in Figure 149.

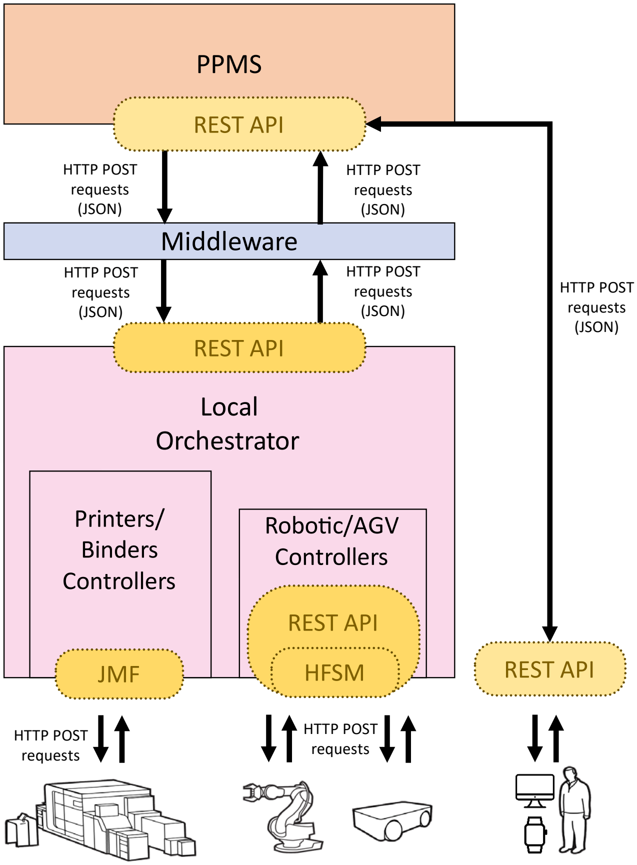


Figure 149: CPS architecture model, developed for CPP pilot. A Printing Process Management System (PPMS) orchestrates, in a global level, the activities of heterogeneous actors, synchronized locally by a Local Orchestrator. Communication between the two levels is performed through middleware.

PPMS takes care of the process advancement. It selects the actors and delivers tasks to them, through the middleware and the Local Orchestrator. The Local Orchestrator communicates with the controllers of the devices, which eventually trigger their action. For instance, it exchanges JMF messages with the printer controllers to initiate printing tasks or request their status. Similarly, it communicates with the controllers of the robotic arm or the AGV, which utilize an Hierarchical Finite State Machine (HFSM) (Yannakakis, 2000) implementation to trigger their physical stepped motions. Tasks for the human operator are delivered on the web-based tasklist application and a smartwatch application. As these components are part of the PPMS implementation (see Figure 133), a direct interface with PPMS enactment service (through REST API) is implemented, bypassing the typical communication through middleware and Local Orchestrator.

7.1.2.3 Physical demonstrator

The resources used in the demonstrator (listed in Section 7.1.2.1) compose a layout of a small printshop. Figure 150 showcases the unloading of media by the collaborative robot arm. Information on which of the two WIP trays is empty is handled by PPMS.

Figure 150: Media unload from printer by a collaborative robot arm[89].

Figure 151 shows the AGV with the motorized arm and a deposit tray, in front of cover printer.

[89] Robotic developments of Figure 150 and Figure 151 were implemented by the EIT OEDIPUS project partner CEA - http://www.cea.fr/

Figure 151: AGV with motorized robot arm and deposit tray for media unloading and transportation, in front of a cover printer.

Figure 152 shows the UI of the smartwatch tasklist application that notifies the operator for pending tasks. Once they perform them, they indicate this to the application, so the workflow advances to the next work item.

Figure 152: Smartwatch tasklist application for human operators.

The PPMS, as a realization of the MPMS architecture model (Figure 130) offers monitoring functionality for a complete production status overview. Two kinds of cockpits were implemented/used, both using the same production status information generated by PPMS. One based on the physical layout of the print shop, i.e., positioning virtually all the production devices/equipment in a nice layout. A pop-up menu list of orders was available for the operator/production manager to point where each order was. Figure 153 shows a snapshot of this cockpit. As can be seen, the selected orderline was in both the CP and BBP. Meanwhile, it is shown that the AGV is moving from the input WIP of the BBP onto the output WIP of the binder to pick-up a bound book from another orderline. Other information, like the time for a printer to get idle, is shown as well.

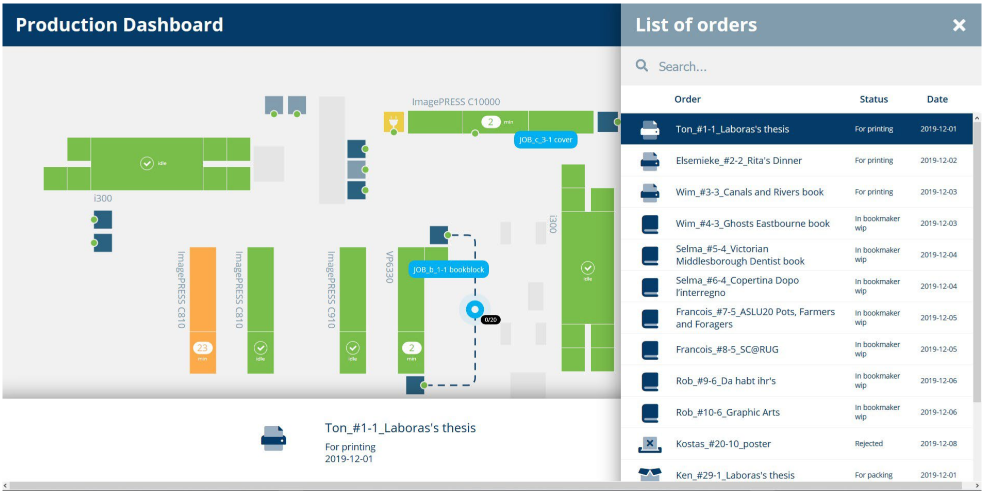


Figure 153: Production cockpit, physical layout view.

Apart from the physical layout view, which visualizes the production status in a more intuitive way, managers may be interested to have a process-oriented view. That is shown in the snapshot of the second type of cockpit in Figure 154 , where the running instances of the process models are listed with their execution information.

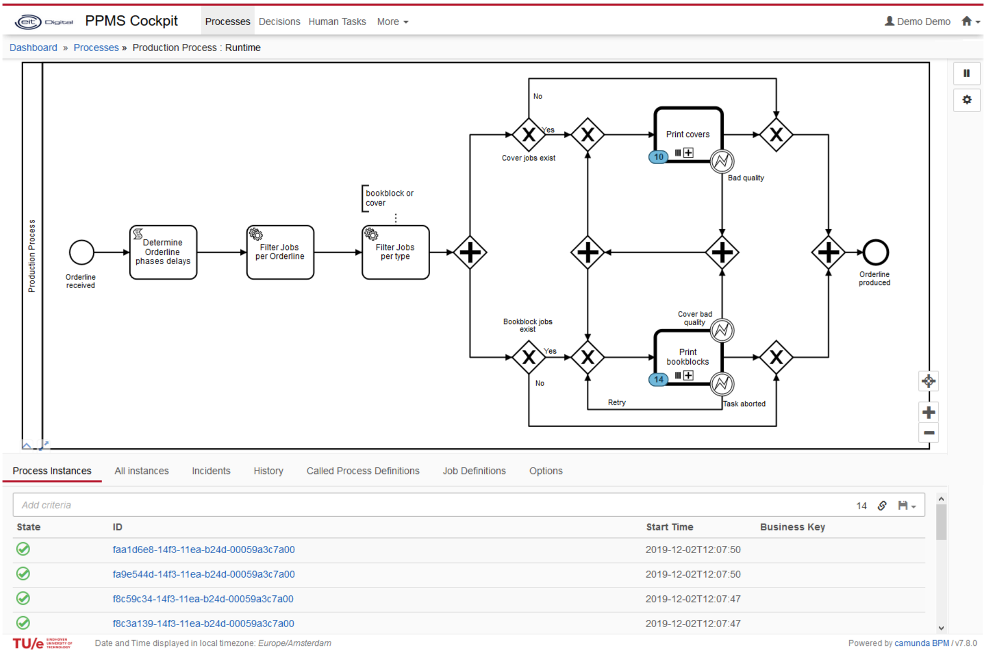


Figure 154: Production cockpit, process view.

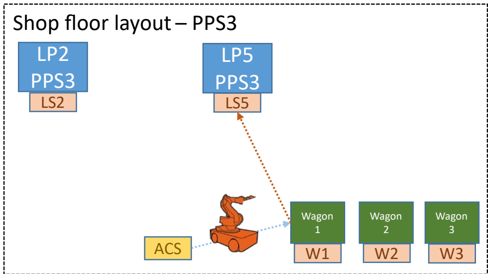
7.1.3 BOS

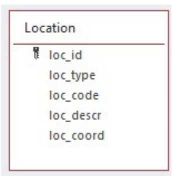
The demonstrator took place in the production environment of BOSCH factory in Madrid, Spain, in the context of the SHOP4CF project.

7.1.3.1 Scenario(s) description

The feeding process of empty trays into the packaging stations currently relies on operators who visually see that empty trays are needed. In the intervention scenario, the physical observations and activities from the operator are heavily delegated to a mobile robot. The robot is an AGV with a robot arm mounted on it. A special gripper is designed to grasp empty trays. As the mobile robot moves across different locations in the production area, it is important to represent those. Figure 155 illustrates the physical layout of the PPS3 area, with two loading stations, three wagons, each with different types of trays, and a charging station. Each location gets a code for easier reference when exchanging transportation tasks messages. The representation of the shopfloor locations is based on the data model of Figure 122.

Upon a trigger from the PLC system of the loading station that empty trays are needed, the AGV moves to the corresponding wagon to pick up the trays, and then loads them to the station. Given that triggers can be raised by both loading stations, at any time, the route of the mobile robot depends on the sequence of tasks it has to perform and its current position.





Location

loc_id	loc_type	loc_code	loc_descr	loc_coord
5	ACS	ACS1	Charging Station 1 for AGV	(10,50,20)
6	LS	LS2	LP2 PPS3 loading station	(20, 20, 40)
7	LS	LS5	LP5 PPS3 loading station	(12, 30, 40)
8	W	W1	Wagon storage 1	(30,67,50)
9	W	W2	Wagon storage 2	(45, 75, 45)
10	W	W3	Wagon storage 3	(56, 46, 56)

Figure 155: Shop floor diagram and corresponding location data model.

The various options to orchestrate the activities of the mobile robot and the human operator have been captured in the BPMN process model of Figure 156. The process is initiated by the first trigger from the PLC. The mobile robot receives tasks in a linear way. When a next trigger is raised while the robot is busy, it is assigned to a human operator. As the operator might be busy in other tasks in other production areas, he might miss the task. This is captured with a task deadline event. In that case, the task is queued for the AGV (according to the task queue management pattern).

The process includes same tasks regardless which loading station raised the trigger for requesting empty trays. The information is passed parametrically (as indicated with the task names in the orange-highlighted tasks). Exception handling functionality takes care to catch issues on the tasks performed by the mobile robot and inform the operator.

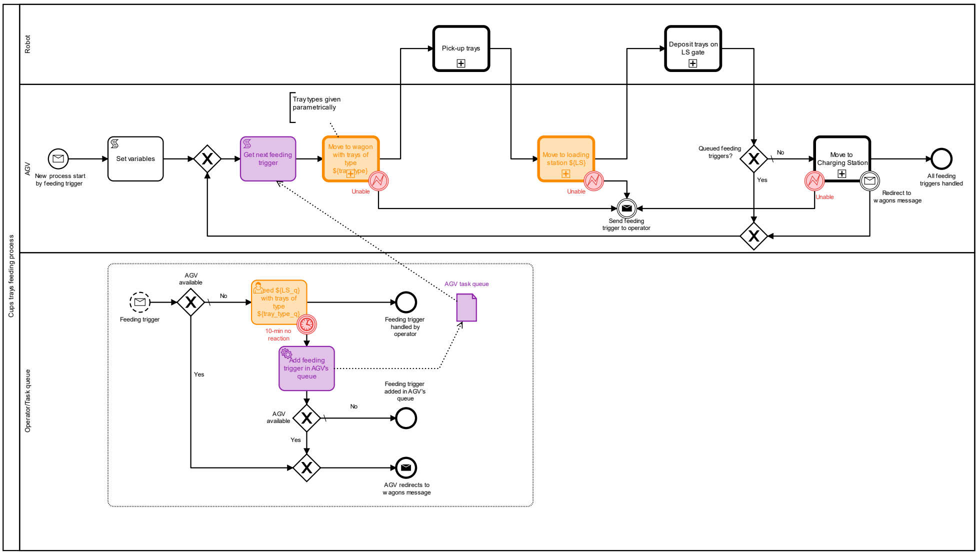


Figure 156: BOS trays feeding process with mobile robot, with queue task management (purple), exception handling (red) and parametrically tasks (orange) to cover both loading stations.

7.1.3.2 Developed CPS

The modeled process is enacted by MPMS. The triggers from the PLC systems requesting empty trays are captured by a Web-of-Things[90] interoperability component, developed by a SHOP4CF project partner. It is translated into an Alert entity[91], posted on the Context Broker of FIWARE middleware. MPMS, which has subscribed to receive such Alerts, handles the workflow by allocating tasks to either the mobile robot (e.g., transportation tasks to the AGV or handling tasks to the robot arm) or the operator. The navigation of the AGV is assisted by a Human-Aware Mobile Robot Navigation (HA-MRN) component, developed by a project partner, to ensure safe navigation on the shop floor (as there are not fixed trajectories and operators might walk along a possible route). The physical devices (AGV and robot arm) are controlled by their controllers.

Figure 157 shows the involved components that constitute the deployed CPS, with a typical sequence of messages exchange. The diagram is based on the SHOP4CF architecture (Zimniewicz, 2020). All the messages between SHOP4CF components (i.e., PLC not considered a SHOP4CF system) are exchanged through the Context Broker for FIWARE middleware, which is omitted in the diagram for sake of simplicity.

---

[90] https://www.w3.org/WoT/

[91] https://shop4cf.github.io/data-models/alert.html

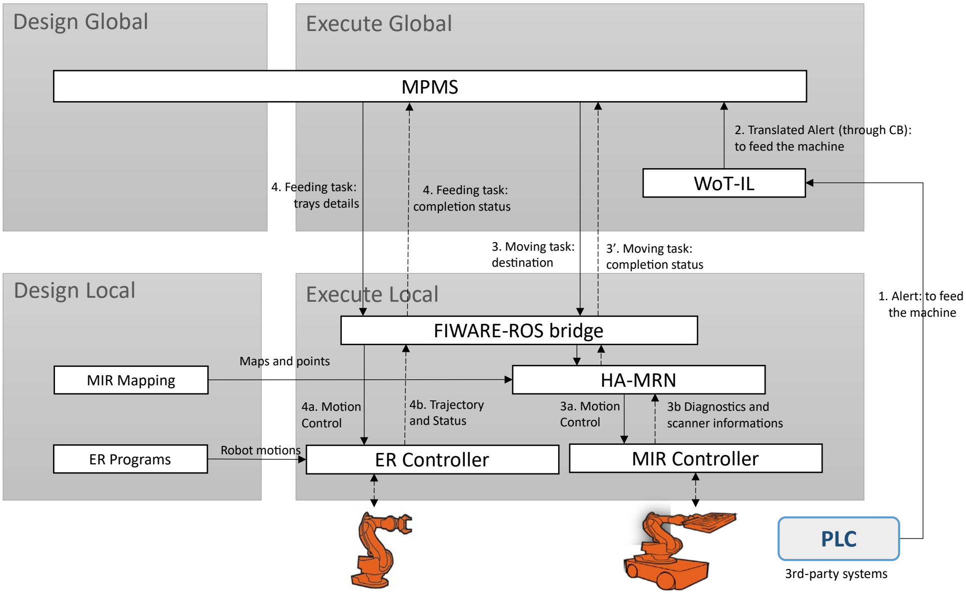


Figure 157: Components diagram as deployed for trays feeding process at BOS.

The execution phase components have been deployed as Docker containers.

7.1.3.3 Physical demonstrator

Figure 158 showcases a snapshot of the mobile robot loading the station empty trays.

Figure 158: Mobile robot (AGV with mounted robot arm with gripper) feeding empty trays on loading station at BOS.

While tasks to the mobile robot are delivered to its corresponding controllers (as task entities through Context Broker), tasks for the operator are delivered through the MPMS Tasklist application. Figure 159 shows a screenshot of the application. It is shown on a screen next to the loading station.

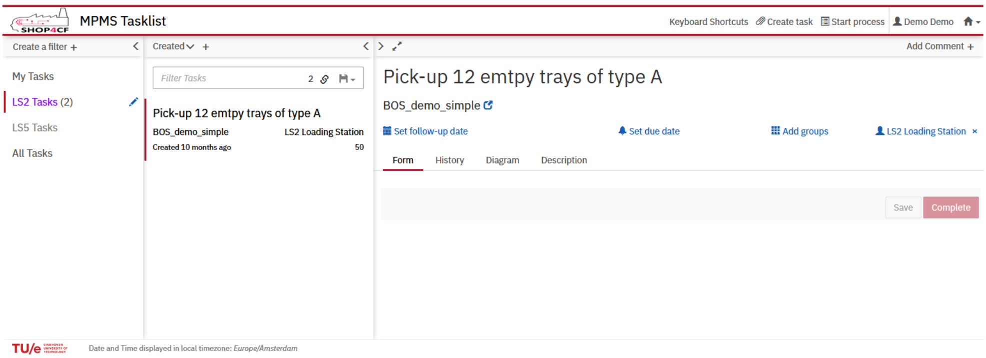


Figure 159: MPMS Tasklist for human operator for the trays feeding process at BOS.

For shopfloor managers who want to get an overview of the activities, MPMS Cockpit application is available, as shown in Figure 160. The application shows the current active tasks and related process information (e.g., which loading station requested which tray type).

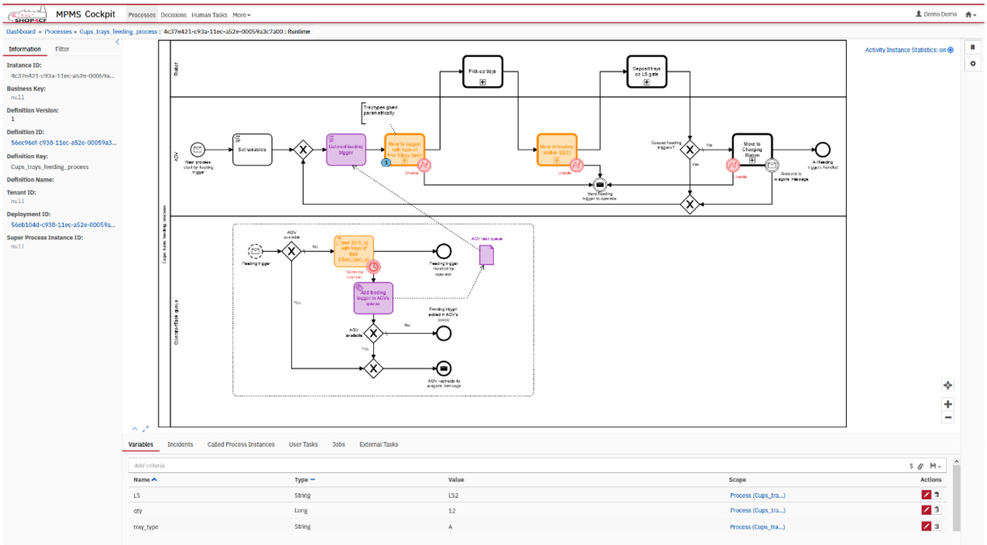


Figure 160: MPMS Cockpit for trays feeding process at BOS. Blue token denotes the state of the process instance. Values of process variables are also available.

7.2 Evaluation

The Framework for Evaluation in Design Science (FEDS) by Venable et al. (2017) has been used to guide the evaluation part of the current research. In their work, they categorize evaluation categories across four dimensions: 1) *why* to evaluate, 2) *when* to evaluate, 3) *how* to evaluate, and 4) *what* to evaluate. The *why*, i.e., the functional purpose, and the *how*, i.e., the evaluation paradigm, compose a two-dimensional framework, as shown in Figure 161. The *why* dimension (*x*-axis) ranges from formative to summative evaluations. The purpose of the former is to help improve the outcomes of the process under evaluation, while the latter aim at judging the extent that the outcomes match expectation. The *how* dimension (*y*-axis) distinguishes between artificial and naturalistic evaluations. Artificial evaluations may be empirical or non-empirical and include laboratory experiments, simulations, mathematical proofs, etc. Naturalistic evaluations explore solutions in their real environments. They are always empirical and typically include case studies, field studies, action research, etc.

As the current research follows the DSR paradigm, and as such its objective is to provide prescriptive knowledge that helps practitioners to solve real problems (as defined in Figure 11), some form of summative evaluation should be performed. The summative evaluation should check utility aspects of the artefacts(s) (as discussed in Section 1.7). Of course, before the final evaluation, various evaluation activities have been performed. The pathway to reach the final evaluation through the intermediate evaluations differs per project. Venable et al. (2017) have identified four main pathways, called evaluation strategies, which are graphically illustrated in Figure 161. Based on the circumstance selection criteria of each strategy, *Technical Risk & Efficacy* has been selected for the current research (highlighted in Figure 161). This strategy is selected:

- if the major design risk is technically oriented, and/or
- if it is prohibitively expensive to evaluate with real users and real systems in the real setting, and/or
- if a critical goal of the evaluation is to rigorously establish that the utility/benefit is due to the artefact, not something else.

The first criterion holds true, especially given the complexity of both MPMS and the CPS in which it is embedded. Regarding the second criterion, placing a solution in production environments is not feasible due to high risks in occurred costs. Instead, sandbox environments are setup to apply solutions in real-world settings. The design(s) shall first be evaluated in artificial environments (e.g., software simulators on development servers) until reaching a mature level to be applied in real-world settings (sandbox). Regarding the third criterion, it is indeed a critical goal to show the impact of the designed solution, regardless how easy or difficult it is to assess this.

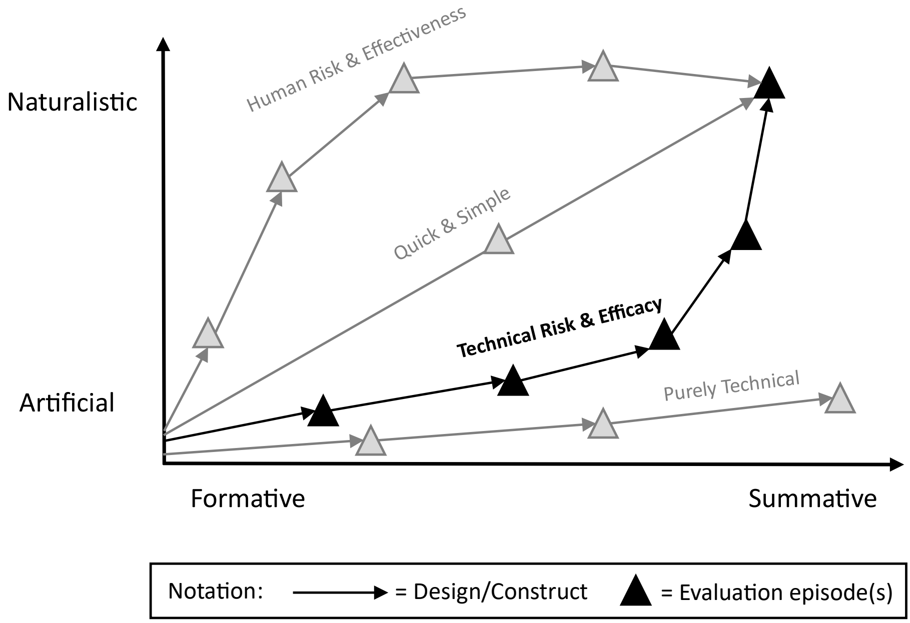


Figure 161: Selected evaluation strategy based on the Framework for Evaluation in Design Science (FEDS) (Venable et al., 2017).

Thus, as this research is a rather long (time-wise) and extensive (content-wise) project, it is evaluated through different evaluation episodes. The more technical evaluation episodes, aiming at verifying the designed artefacts, are discussed in Section 7.2.1. Through the demonstration of the system at various pilots in real-world settings, the final naturalistic and summative evaluation episode is discussed in Section 7.2.2. The focus there is on the user acceptance of the solution(s). Finally, Section 7.2.3 discusses general findings through the demonstration and evaluation efforts.

7.2.1 Verification

The operationalized form of the designed artefacts (presented in Section 6.2) shall be verified to check their correctness, before evaluating their utility. In other words, it should be first checked whether the artefacts work and do what they are meant to do (Gregor & Hevner, 2013; Prat et al., 2015) before checking their value. As all these are finally integrated into an information system, their software constructs are verified based on software quality attributes, according to the ISO/IEC 25010:2011 standard (International Standards Organization, 2011). The main objective is to verify that the proposed solutions address specific issues and requirements, thus, functionality has been selected as the main software attribute. Other attributes such as reliability, efficiency and security are left out of scope given the purposes of this research and the prototypical nature of the implemented solutions.

To check whether the individual solutions function as intended, various test scenarios have been designed and executed throughout the development process, in an iterative form (as the design was also evolving in cycles (Wieringa, 2009)). All the test scenarios follow the same structure: pre-conditions (describing the initial state of the system), steps to be executed, (expected) post-conditions (to be checked with the actual outcomes). For instance, the synchronization mechanism has been verified through automated test scenarios in the form of mock-up recipes, described in the proposed notation as in Table 6. Testing the system with respect to its interaction with other systems, a more integrated approach is necessary, as the one that has been followed in the HORSE project (Arnaudov, 2019). Integrated test scenarios involve the interaction across various components, through which the functionality of MPMS as part of a CPS has been verified.

All the intermediate verification tests raised the confidence for verifying the final system in real-world settings. To verify its functionality, Table 27 discusses how the system requirements (derived from scientific literature and practical analysis and presented in Section 2.3) are satisfied. As a reminder, demonstration has been selected as the primary verification type according to (International Council on Systems Engineering, 2015). As can be seen in the evidence column of Table 27, all requirements have been met, showing that the system is verified with respect to its design. Of course, completeness of the solutions cannot be claimed as the solutions have been tested with respect to specific scenarios. To be more clear, a future scenario which is also considered as a complex one, with respect to R01, but with different setups than the ones verified, might not be addressed with the existing developed modeling solutions. However, considering that the verification scenarios are representative with respect to the high-level system requirements, it is argued that the proven evidence is sufficient to consider the implemented solutions as correct.

Table 27: Verification of advanced MPMS requirements (presented in Section 2.3).

R#	Requirement	Verification type	Verification scenario(s)	Evidence
Design				
R01	The MPMS shall provide modeling support of complex processes that involve synchronization of activities by various actors (including human-	Demonstration	CPP	Recipe system for synchronization of printing and transportation activities.

	robot collaboration scenarios)			
R02	The MPMS shall be able to define manufacturing resources, such that it can determine during execution which resource should perform an activity (linked to R06).	Demonstration	(Erasmus, 2019)	Definition of resources and tasks according to theoretical framework.
R03	The MPMS shall be able to define tasks, such that clear control of activities is provided in both modeling and execution phases.	Demonstration	BOS CPP TRI	Definition of tasks with task input and output parameters in all modeled processes.
R04	The MPMS shall be able to represent the physical equipment hierarchy, such that functional processes are mapped to their respective physical environment.	System realization	BOS CPP TRI	Definition of location (points) to represent physical location points that are included in the physical processes.
Execution				
R05	The MPMS shall be able to enact the modeled processes in an automated way.	Demonstration	BOS CPP TRI	Automated execution of process models by the Process Engine module of MPMS in all scenarios.
R06	The MPMS shall be able to dynamically select and allocate the most suitable resource(s) to tasks, based on task requirements and resource capabilities.	Test	(Erasmus, 2019)	Resource allocation for tool assembly process at TRI.
R07	The MPMS shall be able to send a list of tasks to be performed by each actor in the production environment, for a specific production order.	Demonstration	BOS CPP TRI	Task delivery mechanisms, either through middleware technologies or through tasklist applications, in all executed processes.
R08	The MPMS shall be able to accept notifications from actors in the production environment regarding a change of manufacturing system status, including actors' availability and status.	Demonstration	BOS CPP TRI	As part of CPS, MPMS receives messages from other systems controlling actors through middleware technologies (MPMS has been registered as a websocket message bus client or context consumer).
R09	The MPMS shall be able to receive events regarding changes of the manufacturing system status.	Demonstration	BOS CPP TRI	As part of CPS, MPMS receives events from other systems regarding statuses through middleware technologies (MPMS has been registered as a websocket message bus client or context consumer).
R10	The MPMS shall be able to react on exceptional events that change the status of the manufacturing system	Demonstration	BOS TRI	Implemented process models include the exception handling modeling constructs to react on deviating behavior.

R11	The MPMS shall be able to monitor the status of the manufacturing system during execution of processes.	Demonstration	BOS CPP TRI	MPMS Cockpit application provides monitoring functionality. Additional dashboard for representing production status from a physical view perspective has been developed as well.
General				
R12	The MPMS shall be able to provide administration of processes.	System realization	BOS CPP TRI	MPMS Admin application manages users, processes authorizations and other administrative functionality.
R13	The MPMS shall be able to integrate to other EIS, including ERP/MES.	Demonstration	BOS CPP TRI	As part of CPS, MPMS receives information from other systems through middleware technologies (MPMS has been registered as a websocket message bus client or context consumer).

7.2.2 User acceptance

The advanced MPMS, with its individual developments, has been verified as functioning as intended (with respect to addressing specific requirements). However, it should be assessed whether the end users (i.e., practitioners) accept the proposed concepts and technologies for solving their problems. For this purpose, the Technology Acceptance Model (TAM) (Davis, 1989) is used. TAM is commonly used to explain and predict user acceptance (or rejection) of information systems. According to TAM, three validated scales can measure the perceived ease of use and perceived usefulness to indicate whether users intend to use specific artifact(s).

Semi-structured interviews (Adams, 2015) were held to gather feedback from practitioners on utility of advanced MPMS. This type of interview was chosen to allow for better explanations and follow-up questions, compared to static questionnaires. The structure of each interview was as follows:

1. Introduction, to explain the purpose of the interview and its setup;
2. Background information was collected with respect to participant's job domain and familiarity/expertise on BPM approaches;
3. Brief overview of advanced MPM developments as have been developed within this research and demonstrated within the three European projects. Note that all participants had already seen in practice the proposed solutions as all of them were involved in the design, development and/or deployment of the complete CPS solutions in their respective pilots;
4. Close-end questionnaire to get responses on perceived ease of use, perceived usefulness and intention to use of the advanced MPMS. The questionnaire consisted of a set of statements (based on the evaluation method of Moody (2003), adapted for the specific artefacts that this research has generated), listed in Table 28. For each of the statement, a 5-point Likert scale was used to capture the level of agreement of the interviewees (1= Strongly agree, 5 = Strongly disagree). Note that some statements are deliberately presented in negated/reversed form to keep the

attention of the interviewees high (respectively the results of that statements are interpreted in reversed form);

5. Open-end questionnaire to get more general feedback on the developments.

The evaluation form used in the interviews is available in Appendix L.

Table 28: Evaluation criteria and corresponding statements to measure "utility" aspects of advanced MPMS (on a 5-point agreement scale, ranging from strongly agree to strongly disagree).

	Statement
	Perceived Ease of Use
1.	Modeling processes with MPMS modeler is easy for me.
2.	I find it difficult to provide definitions of involved entities (i.e., resources, tasks, location points) through MPMS.
3.	I find that implementing any business logic (through coding an application project) is difficult for me.
4.	The configuration and customization of MPMS is easy for me.
5.	My interaction with the MPMS applications (Tasklist, Cockpit, Admin) is clear and understandable.
6.	I find it takes a lot of effort to become skillful at using MPMS modules.
7.	I find MPMS rigid and inflexible to use.
8.	Overall, I find MPMS easy to use.
	Perceived Usefulness
9.	Using MPMS (Modeler) allows me to clearly represent (production) processes.
10.	With MPMS (Modeler) it is possible to model complex production scenarios.
11.	MPMS allows for process integration and automation that would be not possible (or difficult to achieve) without this system.
12.	MPMS would enable higher productivity through process management.
13.	With MPMS I get a clear overview of (production) processes.
14.	Overall, I find MPMS useful for tackling process complexity.
	Intention to use
15.	I would consider using the MPMS solutions for tackling process complexity in my organization.
16.	I intend to use MPMS for tackling process complexity in my organization.

Six practitioners from five organizations were interviewed (their responses are added in Appendix M). Their roles and expertise are categorized as listed in Table 29:

Table 29: Profiles of practitioners of evaluation interviews.

Practitioner	Role	Tenure
P1	System integrator	2-4 years
P2	Product/system designer	2-4 years
P3	Workflow architect	>10 years
P4	Workflow architect	>10 years
P5	Managing director	>10 years
P6	Product/system designer	>10 years

Note that all interviewees were/are involved in the three European projects (Section 2.2.1) (they all are members of organizations participating in the respective projects) and have seen and/or applied the developed solutions.

The graph in Figure 162 shows the survey participants' familiarity with the BPM paradigm. While most of them do not consider themselves experts, all expressed their interest on what BPM can bring in their manufacturing operations.

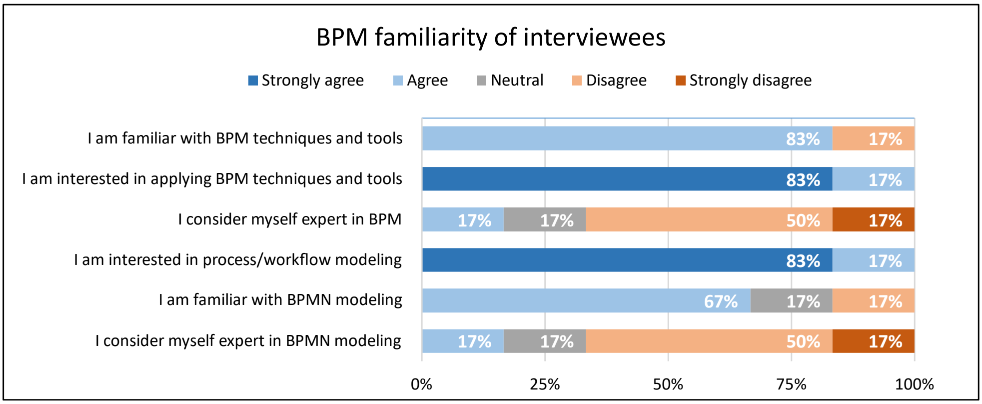


Figure 162: Responses to questionnaire on the familiarity with the BPM paradigm of interviewees that evalued advanced MPMS.

The following three graphs (Figure 163, Figure 164, Figure 165) present the aggregated responses to the close-end questionnaire regarding Perceived Ease of Use (PEoU), Perceived Usefulness (PU) and Intention to Use (ItU) respectively. Statements marked with an asterisk (*) are reversed compared to the statements in the questionnaire to keep homeomorphism of the presentation of the results towards one scale (i.e., easiness).

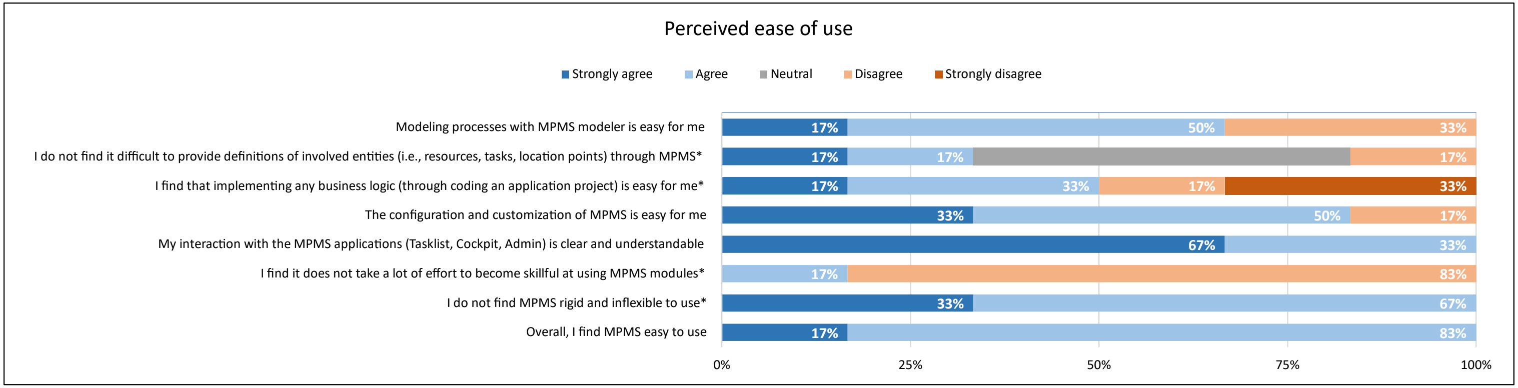


Figure 163: Responses to questionnaire on Perceived Ease of Use (PEoU) of advanced MPMS.

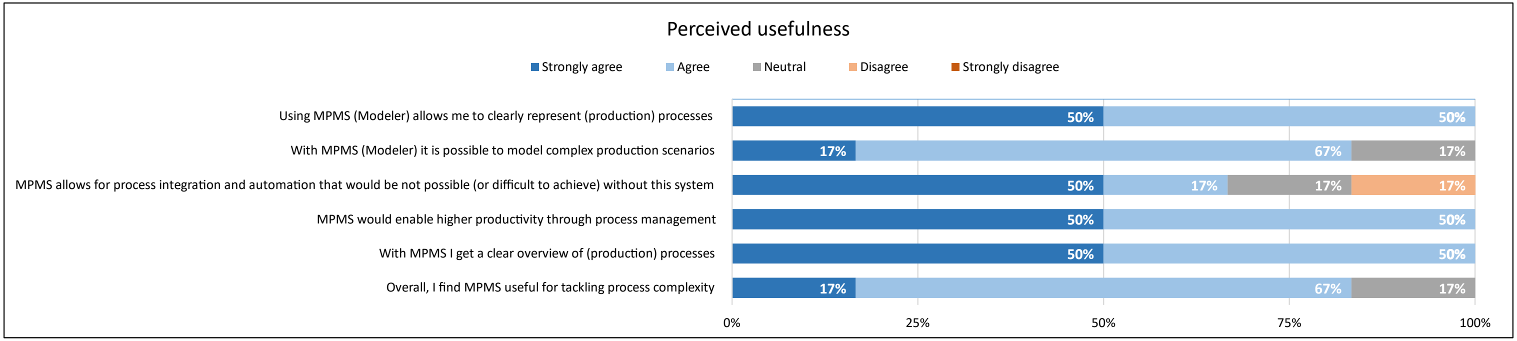


Figure 164: Responses to questionnaire on Perceived Usefulness (PU) of advanced MPMS.

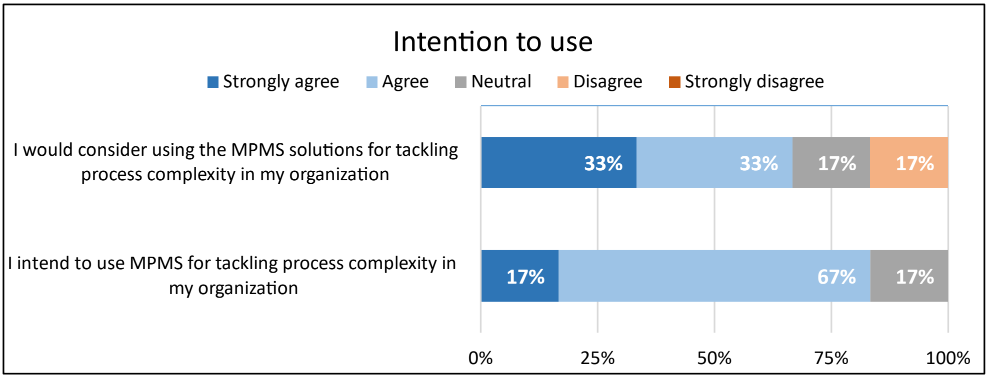


Figure 165: Responses to questionnaire on Intention to Use (ItU) of advanced MPMS.

While the number of participants is rather small, the results show some clear indications on the usability aspects, with insightful points from the justification from the interviewees. The following paragraphs interpret and discuss the results.

<u>Perceived use of use:</u>

- Respondents might have difficulties to model processes, which can be attributed to their low expertise in BPMN.
- The high score in neutral responses with respect to the use of the Support tool for defining involved entities was rather expected. The practitioners did not have much interaction with the tool as at the moment of requirements analysis and designing of the scenarios, the tool was not at its final level.
- The implementation of any business logic of the process models seems to be the hardest part of using the advanced MPMS modules. As this requires coding, the interviewees do not seem confident with this part as this is not their role. Moreover, the selected coding language plays a role, as one interviewee commented that the selected Java language required extra efforts from his side. All respondents, though, indicated that the coding should be doable for a team of developers in their organization.
- The realized MPMS has a rather intuitive interface but there are comments/suggestions for improvements, not only from a user-friendly perspective (e.g., showing informational messages in some cases), but also from a functional perspective. For instance, the Cockpit application could be more interactive when managers could select and view more details of a production order/instance.
- All interviewees indicated that it would take considerable effort to use all advanced MPMS modules. This can be attributed both to their non-high expertise in the concepts and that each module requires different skills.
- Respondents seem confident that with the designed modeling constructs would be able to model complex production scenarios. However, they indicated that their response should be treated with caution as there might be complex scenarios that their organization does not include, and thus, they cannot predict how useful the existing constructs would be. This relates to the observation made in Section 7.2.1 with respect to the completeness of the verification evidence (cf. Table 27).

Perceived usefulness:

- The usefulness of MPMS for process automation and integration received positive reactions. In that respect, a process orchestration system is essential in organization. However, as a few respondents commented, they cannot attribute all the merits to MPMS as they would like to explore alternatives. This is a fair comment, especially for SMEs which might not employ at all a (similar) system for process management.
- The process automation, the integration of systems, and the orchestration of actors that the advanced MPMS offers are definitely enablers for higher productivity. Of course, as many other factors might affect productivity (e.g., type of technologies), it is hard to judge (especially without quantitative metrics) what is the effect of the system.
- A similar explanation is given for the overall question whether MPMS helps in tackling process complexity. The proposed solutions are deemed promising on that direction, but there are more aspects to consider regarding complexity issues in organization (e.g., the size of the enterprise information systems landscape that MPMS has to fit in, the maturity level of the solutions, etc.).

Intention to use:

- Practitioners seem willing to use the advanced MPMS, or at least part of the developments (e.g., modeling). However, the actual usage might be affected by factors such as competitor systems (either within the organization or offered by commercial providers), fit in the entire IS landscape, or existing level of expertise to support it.

7.2.3 Findings

The demonstration of advanced MPMS is deemed successful, verifying that the system design satisfies its purposes to solve practical needs. The evaluation survey on user acceptance of the advanced MPMS technologies through interviews with practitioners shows promising results on the utility of the system. The demonstration efforts and the discussions with practitioners raised, though, a few interesting points, which are discussed in the following paragraphs.

While most of the demonstrators and pilot scenarios had a rather small scope, the essence of using the advanced MPMS as a process orchestrator for activities performed by heterogeneous actors became apparent in most of the cases. For instance, using a mobile robot to serve one production line during a demonstrator might make someone think that this could be automated by any software that handles the device(s). But when the organization wants to scale up such solutions in many production lines or using many more devices, a process overview is missing. Thus, while a software system that manages a (fleet of) mobile robots can serve well the activities of these devices, such a system is unable to manage a set of heterogenous actors that need to perform similar activities or collaborate with each other. Think for instance the BOS pilot case (Section 2.2.4) in which either a mobile robot or a human operator can perform loading/transportation activities. A system like MPMS is necessary to orchestrate their activities. Of course, large organizations might have a system (e.g., an MES with process management functionality) to provide a process perspective, but for SMEs the lack of such systems is impeding for embracing new technologies.

MPMS is a multi-faceted information system with various modules dealing with both design and execution phases (Figure 130). Each of the modules shall be used by different roles, so different expertise is required. More specifically, a process modeler will use the Definitions Tools modules, a developer will implement any advanced business logic of the process models (i.e., developing the Core application), a production manager will configure processes and resources on Administration Tools and monitor those on Monitoring Tools. The researcher of this thesis had/acquired the knowledge to deal with all modules, however, this might not be the case in organizations. Various people shall be responsible in employing and using MPMS. This observation was also obvious during the evaluation sessions, in which practitioners did not have the same level of confidence to judge the various components. Consequently, the evaluation results should be treated with caution.

Regarding the process modeling, the current research proposes BPMN, as a modeling language which is highly expressive, entails execution semantics for automated process enactment, is promising for business and manufacturing processes integration, and can be easily adopted as an open-source standard with strong academic and commercial support. The designed developments (i.e., artifacts for flexible process modeling and exception handling) enrich the language for adoption in smart manufacturing. However, as many legacy and bespoke systems in organizations typically use their proprietary modeling languages, it might be hard for BPMN to find its position in the manufacturing process modeling domain. This, though, does not invalidate the existing efforts and should not discourage further developments, especially when there is high interest to provide a process modeling perspective that organizations miss.

The unified process management approach, that the current research proposes, distilled in a single process management system, has to reach a broad application in practice to prove its promises. Especially when Industry 4.0 demands for decentralization, a centralized system such as MPMS might be proved to be cumbersome. However, the current research has put forward approaches that might be missing in practice at the moment.

7.3 Chapter conclusion

The application and demonstration of advanced MPMS developments, through complete CPS solutions, proved feasibility of the designed artefacts. In a substantial set of real-world scenarios, practitioners were able with the MPMS to:

1. Model complex production scenarios, e.g., mobile robot sharing across loading stations or synchronized media unloading from a printer by a mobile robot.
2. Handle operational exceptions, e.g., machine failures by transferring tasks to human operators.
3. Orchestrate the activities of heterogeneous actors (through vertical control) in a cross-level perspective (horizontal integration).

The developments, having as objective the generation of utility, were evaluated in the context of the pilot cases with respect to perceived ease of use, perceived usefulness and intention to use. Practitioners are positive on applying the proposed solutions in their organizations, as they find them useful towards tackling their operations challenges. While they posed some concerns regarding the easiness and flexibility on using a centralized process management system, they recognized the need for a process perspective in their organizations.

CHAPTER 8

Conclusion

The manufacturing industry is going through disruptive changes, both from business and technology perspectives. Market trends such as mass personalization and high fluctuation in demand for material and products compel manufacturers to seek for flexibility in their operations. Product variety imposes production and equipment variety (Brunoe & Nielsen, 2016; Johansson et al., 2016), often causing increased complexity of production operations (ElMaraghy et al., 2013; Hu et al., 2011). Dynamic market environments increase uncertainty and demand high responsiveness not only on strategic or tactical level but also on operational level. On the other hand, the rapid technology developments with advanced robotics, augmented reality systems, and automated guided vehicles, all leveraged by the connectivity of IoT and cloud computing, promise for increased productivity, higher efficiency, flexibility and labor cost reduction (Dalenogare et al., 2018; Hofmann & Rüsch, 2017). However, the transition from a traditional factory into a smart one is typically achieved in stages, resulting, often, in isolated, fragmented developments that do not solve the need for production adaptability and flexibility.

All changes result in a general complexity problem in manufacturing operations, impeding manufacturers, especially SMEs, to harvest the Industry 4.0 benefits. The research presented in this thesis aims to provide support in tackling operations complexity with process management theories and techniques. The identified general problem is decomposed into specific challenges, for which solutions are designed and developed.

This last chapter summarizes and concludes the research presented in this thesis. Section 8.1 presents a summary of the research and reflects on how the various developments fulfil the research objective(s). Section 8.2 describes the contributions, both to research and practice. Section 8.3 discusses the limitations, followed by a few research directions for future work in Section 8.4. Finally, Section 8.5 concludes this thesis with final remarks and takeaway messages.

8.1 Research summary

Design science research approach (Hevner et al., 2004; Peffers et al., 2007) is followed to provide useful solution(s) to the identified problems and challenges. Extensive problem analysis both from scientific literature and practical perspectives (as elaborated in Chapter 2) found three main categories of causes that lead to the general operations complexity problem in smart manufacturing:

- Complex production scenarios (due to both market push and technology push);
- Rise of exceptions (due to both market dynamicity and increasing used technologies);

- Integration complexity (due to technologies heterogeneity and fragmented solutions).

It is argued that BPM theories and tools can provide support in the abovementioned identified challenges. While the paradigm has proven its strength in various domains, its application in manufacturing is not mature, yet receiving a lot of research interest (e.g., (Pauker et al., 2018; Prades et al., 2013; Schönig et al., 2018; Zor et al., 2011)). Thus, the research presented in this thesis is dedicated to applying and extending BPM in smart manufacturing with the main research objective *to provide models/constructs, guidelines and specifications of systems to apply advanced process management in smart manufacturing to tackle process complexity*.

To accomplish the research objective, the following main research question has to be answered:

RQ: How can manufacturers tackle the process complexity in dynamic, discrete, smart production environments, in terms of flexible modeling and responsive enactment of their processes?

To provide an answer to the above research question, four sub questions have been defined (as introduced in Chapter 1). The following paragraphs discuss how these research questions have been addressed.

RQ1: How can we provide flexible modeling of complex production processes?

Production processes are getting more complex, not only due to process variety to provide the demanded product variety, but also due to the different actors that perform activities and need to collaborate or synchronize with each other. In the line of applying BPM in smart manufacturing, BPMN as the de facto standard for business process modeling has been selected to cover the modeling of manufacturing processes. BPMN has been selected based both on the increasing interest for its application in manufacturing (Section 3.2) and the work of Erasmus et al. (2020), which investigates the completeness and suitability of BPMN 2.0 for representing manufacturing operations. That work has to be extended to cover more detailed modeling requirements and be automatically enacted.

Analysis from practice (i.e., the intervention scenarios from projects' pilots) had shown that there should be modeling support for: i) task delivery to heterogeneous agents, i.e., how work instructions are delivered to the actors (either human or automated operators) that perform activities, ii) human-robot collaboration, i.e., the interaction between a human operator and a robotic device/equipment on performing required operations, and iii) the activities synchronization, i.e., the points in time and space dimensions at which two or more activities have to be synched. The first two requirements have been addressed with the design of BPMN modeling patterns/constructs. The third one has been addressed with a synchronisation mechanism, called recipe system. The recipe system has been designed and formally described to provide modeling support for common manufacturing constructs, namely buffering and (un)bundling.

RQ1 has been addressed in Chapter 3.

RQ2: How can events and exceptions be handled in dynamic manufacturing environments?

The rise of exceptions requires manufacturers to be responsive to avoid downtimes, extra costs and/or performance deterioration. The first step to handle exceptions is to have a clear picture of the type of exceptions. A categorization of exceptions has been constructed from both input from literature and practice. An SLR has identified type of exceptions in the manufacturing domain that appear in literature. Input from practice has been gathered to verify literature and identify additional type of exceptions that occur in practice. The input was both in the form of an empirical database containing information on exceptions (quantitative) and through interviews with practitioners on what exceptions occur in their organizations (qualitative).

With a categorization of exceptions, the determination of a suitable handling strategy was the next step, as corrective action(s) depend on the type of occurred exception(s). A set of decision trees have been designed that guide the selection of handling strategies. The guidelines take also into account KPIs, as the performance of the organization might affect its corrective mechanisms.

RQ2 has been addressed in Chapter 4.

RQ3: How can we enable process integration for end-to-end manufacturing process management?

Process management in enterprises is often fragmented, with different techniques applied in different parts of the enterprise. A broader, cross-functional overview is missing, hindering flexibility. A single process management system is required to unify process management functionality offered by traditional systems such as ERP and MES (Erasmus et al., 2018). Such a system has been designed based on traditional BPM systems, called MPMS.

The logical view of MPMS (i.e., its functionality) has been described with an updated version of the 5-aspect framework of Truijens (Grefen, 2016). The data aspect is covered with the design of relevant concept data models. The software aspect is covered with the design of a logical software architecture, based on three reference architecture models for BPMS. More specifically, the long-established Workflow Reference Model (WfRM) of the Workflow Management Coalition (WfMC) (Hollingsworth, 1995), the Mercurius reference architecture (Grefen & Remmerts De Vries, 1998) and the novel BPMS reference architecture (BPMS-RA) (Pourmirza et al., 2019) have been considered. The software architecture includes logical modules for both design and execution phases of manufacturing operations.

The conceptual design of the system is complemented with specifying the interfaces of MPMS to other systems. This is achieved by positioning MPMS in a CPS. The reference architecture of the HORSE CPS system (Grefen & Boultadakis, 2021) has been selected as a basis for that reason. The HORSE system is a modular architecture for integrated manufacturing process management of heterogeneous advanced technologies.

The design of MPMS specification has followed an iterative process (Wieringa, 2009) with various refinements towards the presented final result.

RQ3 has been addressed in Chapter 5.

RQ4: How can an advanced manufacturing process management system support the complexity tackling in smart manufacturing environments?

In order for MPMS to support all three identified aspects of complexity, it has to include the required functionality to do so. That means that the system should include operational support of the conceptual designed artefacts for flexible process modeling and exception handling. Moreover, the conceptual specification of MPMS has to be further described from a development point of view. Thus, an architecture model of an advanced MPMS has been designed, as the ensemble of the operationalized three designed artefacts. To prove feasibility of such an architecture, a system realization with concrete technologies has been presented as well.

RQ4 has been addressed in Chapter 6.

8.2 Contributions

The design science research paradigm, which the current research has followed, aims at providing knowledge to address practical needs/problems (Hevner et al., 2004). The knowledge is in the forms of artefacts, which are designed and developed with the goal to generate or improve utility, i.e., create artefacts that are useful and purposeful to solve business problems. While the artefacts can be readily available for application (either directly or through concrete instantiations in case of abstract artefacts), their constructions also generate contributions that extend the knowledge. The extended knowledge can be then used to solve new or adapted business needs.

Accordingly, the research presented in this thesis has generated knowledge contributions. It is concerned with the application and extension of BPM for tackling process complexity issues in smart manufacturing. According to the DSR knowledge contribution framework of Gregor & Hevner (2013), the research is positioned at the intersection of invention and exaptation, as discussed in Chapter 1. The following two subsections discuss the significant relevance and value of knowledge contributions, both to research and practice.

8.2.1 Scientific contributions

As the general process complexity problem has been decomposed into individual problems, a set of artefacts has been generated to provide support in all individual problems. As the artefacts are in the form of constructs, methods, guidelines and (architecture) models to be applied by practitioners, they constitute prescriptive knowledge (in contrast to descriptive knowledge). More specifically, the following artefacts have been generated:

1. A set of modeling constructs to represent (complex) manufacturing operations processes.
    a. Task delivery patterns;
    b. Human-robot collaboration patterns;
    c. (Activities) Synchronization mechanism.
2. A categorization of exception types appearing is smart production environments and set of guidelines to determine suitable handling approaches.
3. A specification of an information system to design and enact manufacturing processes, as part of a CPS.

4. An architecture model of an advanced manufacturing process management system that integrates the first three design artefacts.

The theory of invention and exaptation of BPM for smart manufacturing operations management is comprised of all the above artefacts. Complementing the work of Erasmus (2019), which focuses on dynamic resource allocation, the theory presented in this thesis extends the reach of the BPM paradigm in the manufacturing domain. While BPM has seen interest and has been applied in manufacturing, the current research applies the paradigm to a wide extent, covering many aspects (e.g., modeling, inclusion of resources/agents, exception handling, technologies/systems to support the runtime execution, integration to other systems and process monitoring). This is in comparison to other works that partially apply BPM theories and techniques, as has been discussed throughout the thesis and summarized in Section 2.1.7.4 (e.g., approaches that focus only on process modeling with BPMN extensions without execution support or others which provide also execution engines to enact process models but do not (extensively) cover aspects such as exception handling or process monitoring). Moreover, the current research puts focus on integration aspects by placing MPMS within the context of a CPS system, to enable and realize the horizontal and vertical integration that Industry 4.0 demands (Kagermann et al., 2013). In that respect, the architecture model of the advanced MPMS contributes on how BPM systems could be realized for application in smart manufacturing. With respect to modeling of manufacturing processes, the proposed modeling constructs have been designed to cover as many manufacturing operations scenarios as possible. This enriches the already powerful BPMN language and contributes to its use in physical domains. Especially, the synchronization mechanism provides BPMN support for common manufacturing constructs (i.e., buffering and (unbundling)) that the notation inherently lacks (as discussed in Section 3.5.1). With respect to exception handling, the designed categorizations and handling guidelines, together with their modeling (and execution) support, provide a compact overview of how BPM approaches should treat exception handling in physical domains such as smart manufacturing. That is a clear contribution compared to works which either deal with exception handling in manufacturing without explicit process support (e.g., (Keddis et al., 2016)) or provide process support but without (clearly) addressing smart manufacturing characteristics (e.g., (Reichert & Weber, 2012a)). Finally, all the developed solutions have been applied in real-world settings (and not remaining on theoretical level), through prototype demonstrations, proving feasibility. Through the application and evaluation, valuable insights have been gained, as further discussed in the next section.

8.2.2 Practical contributions

As already stated, DSR targets to solve problems rooted in practice. Practitioners do not always care about how solutions are designed but whether these eventually address their needs/problems. The current research has created knowledge to address the identified problems in the following ways:

- With the use of BPMN, which covers also complex production scenarios through the developed modeling constructs, practitioners have an expressive way to model scenarios, which can also be enacted during runtime (compared to other notations which do not have execution semantics). This is useful especially for manufacturers which do not adopt process modeling (while their processes are getting more complex), or they do it in an ad hoc way, or they are limited to proprietary modeling languages. Moreover, the use of BPMN for both business and manufacturing processes can facilitate process integration between Level 4 and Level 3 of the IEC 62264-1 standard functional hierarchy, considering that the notation is already the de-facto standard for business process modeling.

- Through a clear categorization of exceptions and a set of handling guidelines, practitioners have a structured way to handle deviations at operational level. Operators, even with low experience or knowledge, can follow clear instructions on selecting appropriate handling strategies. Providing also automated exception handling support during execution of processes, enables organizations to be more efficient and responsive in dynamic environments, with, possibly, reducing negative effects such as high downtimes, production errors or incurred costs.
- A process management information system enables cross-functional process integration and automated execution of modeled processes in which heterogeneous actors are involved. As has also been indicated by practitioners during the evaluation sessions, an end-to-end process view is often missing, exactly due to the lack of a process orchestration hub. Especially for SMEs which do not have the resources to afford a commercial system with process management functionality that can play that role, MPMS can be used to enable horizontal and vertical integration, as it has been designed to be part of a complete CPS.

The practical significance of the realized advanced MPMS is further demonstrated through three European projects (introduced in Section 2.2.1), in which fourteen (at the moment of writing this thesis) pilots operating in various sectors have applied and evaluated it (see Appendix K). MPMS has been a core component in the HORSE CPS framework (Traganos et al., 2021), which jointly with middleware and robot control systems provides seamless integration across the entire functional hierarchy. Ten pilots in the HORSE project used MPMS to solve their needs (as briefly listed in Appendix B), getting introduced to BPM approaches. In the EIT OEDIPUS project (Traganos, Vanderfeesten, et al., 2020), BPM has been introduced to enable automated production printing, in a domain where the paradigm was not applied before (at least to that extent). Similarly, in the SHOP4CF project, various pilots have applied developments that the current research has generated to orchestrate the activities of the robotic solutions they aim to introduce in their factories.

Thus, manufacturing process management, as the current research proposes it, has reached practice which now (re)considers of how operations are orchestrated.

8.3 Limitations

The demonstration of advanced MPMS developments has proven application feasibility and the positive feedback through the evaluation indicates that the current research has made contributions in the right direction towards solving the identified challenges. However, the research has a few limitations which are discussed in the following paragraphs.

During the analysis phase for identifying problems and setting the requirements for the solutions, practical relevance has been ensured by investigating challenges and Industry 4.0 endeavors in real factories. While the European projects, in which the author has been involved, provided a reasonable set of pilots with various scenarios, there is risk of overlooked problems. Scientific literature has been extensively consulted but, as the manufacturing domain is rather vast, diverse and fast evolving, the current research might have missed to cover existing problems. For instance, at the moment of writing this thesis, no pilot that had been analyzed used any virtual reality (VR) technologies (which are gaining a lot of interest (Zhou et al., 2019)). It would be interesting to explore how MPMS could deliver tasks to human operators that use VR devices or how production status (currently available in dashboards) could be interactively visualized in augmented technologies.

The current research is practice-oriented and, thus, less emphasis has been put on formalizing all conceptual designs. Apart from the synchronization mechanism (i.e., recipe system), which is a novel approach to cover an inherent limitation of the BPMN language (i.e., process instances correlation), other less complex constructs have been designed with practical applicability as the main drive. For instance, the task queue management pattern (Figure 60) could have been formalized with queuing theory (e.g., representing the alternative task paths as queuing buffers, denoting whether they refer to a single task or a series of tasks). Similarly, the various task status values that are used in task exception handling could have also been defined as a state diagram, complementing the designed BPMN model construct (Figure 118). The lack of such formalizations, though, does not invalidate the scientific rigor of the artefacts as these are constructed based on extensive requirement analysis and clear design steps.

As has already been mentioned in the evaluation discussion (Section 7.2), the final naturalistic and summative evaluation has been performed on the realized advanced MPMS and not on the architecture model itself. Reference software architectures are typically evaluated on their quality (Angelov et al., 2012; Dobrica & Niemelá, 2002) (e.g., assessing maintainability, modifiability, etc.) through scenario-based methods (Babar & Gorton, 2004). Well-known methods are the Scenario-based Architecture Analysis Method (SAAM) (Kazman et al., 1994) and the Architecture Trade-off Analysis Method (ATAM) (Kazman et al., 1998). Scenario-based methods involve various stakeholders (especially software architects), should be performed at different stages of a project, and require tool-support, which few of them provide (Shanmugapriya & Suresh, 2012). It would be hard to perform such evaluations within the scope of the current research. Especially for BPMS architectures, (Pourmirza et al., 2017) found that none of the selected primary studies had evaluated their architectures based on well-known evaluation methods (e.g., SAAM, TAM). Instead, the architectures were evaluated through actual implementation or through case studies for measuring aspects such as evolution or performance. That means that the lack of evaluation guidance would make the evaluation of MPMS harder within the scope of the current research.

The advanced MPMS, as a result of a research project, has been built as prototypical solution (as part of CPS solutions within research and innovation projects) to be validated in real production environments. As such, it is considered at technology readiness level 6 (Mankins, 2009; Olechowski et al., 2015). Since the emphasis was mainly on testing and proving intended functionality, other aspects such as robustness, scalability, user-friendly interfaces, and security, were out of main focus. Any criticism from the pilots on these aspects, should therefore be treated with caution for assessing the adoption of the developed solutions.

8.4 Prospects

The current research provides a solid theory of the application of BPM in smart manufacturing for tackling process complexity. The following paragraphs discuss research opportunities that can further extend the theory.

With respect to modeling, the developed constructs have been designed to cover as many production scenarios as possible, but their utilization will be enhanced once overcoming their design limitations. More specifically, current assumptions in the definition of the synchronization mechanism, such as that pools have infinite capacity or their cardinality is only expressed in units of process instances, can be relaxed. Think again the example of printed media grasping by the mobile robot in CPP pilot (see Figure 151), where the books/covers are represented with the same unit (i.e., as instances of their respective process model), regardless of their thickness and size. Workaround solutions exist, such as using an external knapsack problem solving engine, which passes group information to the recipe system in the form of selector attribute values, so that the recipe system can perform the appropriate bundling operations. Similarly, the assumption that a selector attribute is shared across all pools should be addressed by giving unique object identifiers to each pool. Furthermore, to make the system more dynamic and flexible, the recipes should be (re)configured during runtime and the fulfilment conditions should be variable instead of static. Moreover, new extension BPMN elements can be designed to help modelers identify and represent synchronization points more easily (as currently only vanilla BPMN 2.0 standard elements are used). One such extension can be the use of a "buffer" element, as the one proposed by Aspridou (2017), complementing it with execution semantics. The visualization of BPMN elements for representing manufacturing processes can also be enhanced with new BPMN task types to represent the ones executed by automated actors (instead of using the User or Service tasks). Such visualization extensions, closer to the physical manufacturing world, will hopefully increase the adoption of BPMN in the domain.

BPMN is a very rich language in describing processes but, in some cases, the rather linear approach might be cumbersome to represent all scenarios. There might be scenarios where a sequence of activities is not always clear/predefined or the number of production paths might result in spaghetti-like process models (van der Aalst, 2012). In those cases, a different approach might be more suitable, for instance modeling activities with the guard-stage-milestone (GSM) approach (as discussed in Section 3.2.3). CMMN is a candidate notation to be explored for modeling process modeling. It can also be combined with BPMN, in a hybrid approach (Traganos & Grefen, 2015).

Regarding the logical software architecture of MPMS, which currently considers separation of concerns for Design/Execution phases (per *life cycle & value stream* dimension of RAMI 4.0 framework, as discussed in Section 5.3.1), there should be an extension to include an Analysis phase. Running systems, robots, and devices produce a lot of data that can be useful both in short and long-term for optimization of processes. Similarly, the plethora of events and exceptions should be analysed to find any correlations that might affect (positively or negatively in terms of performance) the operations. Therefore, the Analysis phase shall include components to provide support for analysis of data for optimization of either Execution (i.e., without explicit redesign) or Design (i.e., with explicit redesign). Such an approach is followed in the SHOP4CF project (Zimniewicz, 2020), however MPMS has not covered yet the analysis phase of the project's software architecture at the moment of writing this thesis. The analysis functionality shall conform to the BPI&BPA component of the BPMS-RA.

As discussed in Section 5.2, MPMS has been proposed as an orchestration hub that unifies process management functionality offered by other traditional enterprise systems (e.g., ERP, MES). MPMS is then responsible for process control, with an end-to-end process orientation. On the other hand, business or shop-floor planning has a resource or function orientation. Consequently, there is a gap between process control and planning. In the prototypical implementations in pilots, the planning aspect was not extensively considered (e.g., handling of a set of orders was performed with mock-up planning scenarios). However, further research should address how MPMS (control system) should interface planning systems. There should be clear roles on whether MPMS shall trigger a planning system (Marrella, 2019) or whether a planning system with detailed scheduling shall trigger MPMS to perform executions.

In all pilots' demonstrations, MPMS (embedded in CPS) has covered orchestration and automation of manufacturing processes within the physical boundaries of a single organization or even a single site. However, nowadays, processes for manufacturing products have an inter-organizational scope, where they span across multiple locations or even different, collaborating enterprises, in a complete manufacturing chain or even manufacturing networks. Cross-organizational manufacturing was the main concept of the CrossWork project (Grefen, Eshuis, et al., 2009; Grefen, Mehandjiev, et al., 2009), in which a centralized global business process controls the activities of local processes of multiple organizations. The same approach can be applied in CPS architectures (e.g., the HORSE CPS) where a global process of a manufacturing organization, orchestrated by the MPMS, can synchronize local processes, both within the same organization (but e.g., in different locations) and across other organizations. Such an approach is illustrated with an example in Figure 166. And of course, the collaboration among enterprises does not stay only on the business level (Level 4 of the IEC 62264-1 standard functional hierarchy) but goes in operational and actual control levels through the vertical integration that a CPS offers. This networked process management (Grefen, 2013) has gained increasing attention in Industry 4.0 as well (Schulte et al., 2012; Weyer et al., 2015), especially due to technologies such as cloud computing. An inter-organizational approach can lead to full automation in complex manufacturing network environments. Note that where necessary for security or privacy issues, the management of the local and cross-organizational processes can be handled by separate systems, as suggested by the CrossWork framework.

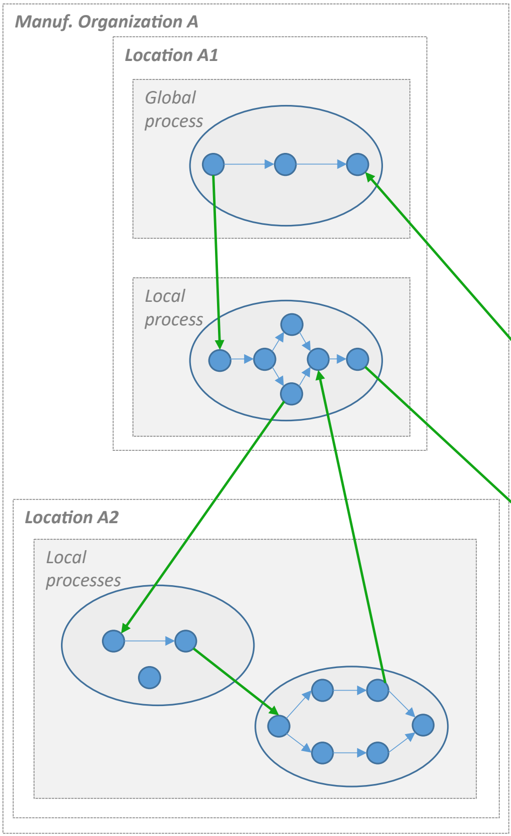


Figure 166: Cross-organizational, networked manufacturing (inspired by (Grefen, Mehandjiev, et al., 2009)).

SMEs, as MPMS targets, often do not deploy robust information systems solutions (e.g., a commercial MES) or high-end control systems. Moreover, there is often limited usage of advanced robotics, due to lack of expertise and/or financial resources, and thus, full automation is not there yet. This limited availability of on-premise infrastructure and computing resources can be addressed with cloud computing technologies (Shawish & Salama, 2014). This means that MPMS (and the CPS in which it is embedded) should support the cloud services paradigm (as currently the solutions were deployed locally, on-sites' premises). The application of cloud solutions may not influence the logical view of the software architectures, in terms of how the designed modules function, but it will definitely influence the physical view (of the Kruchten framework), in terms of how the system will be deployed to exploit computing infrastructure investments. Performance and timing constraints are crucial factors in deciding which services can be brought to the cloud. Modules that require sub-second response times (especially local execution ones) should be better deployed on-premise (local) infrastructures, probably combined with fog computing (Bonomi et al., 2012). On the contrary, modules used during the design phase of processes or modules that are not heavily impacted by internet traffic and communication delays can be hosted as cloud applications, either as Software-as-a-Service (SaaS) or Platform-as-a-Service (PaaS) solutions (Erasmus, Grefen, et al., 2018). The use of cloud computing can also further enable cross-organizational manufacturing as discussed in the previous paragraph (Hans et al., 2013; Schulte et al., 2014).

Production environments are transforming with the introduction of new technologies. Especially for shopfloor human operators, the changes pose new ways of working. They have to use new robots, wear handheld equipment, interact with new HMI devices and operate new software systems. Clear understandability of the functionality and the actual usage of the new systems and technologies is important to ensure that these will be well adopted and proved valuable. As also indicated through the evaluation and feedback discussions, clear instructions and (error) messages shall be presented to operators, who often have to act under time-pressure. Specifically for MPMS, tasks delivered to operators and information shown on dashboards (e.g., running process models) should be clear and non-ambiguous. While user-friendliness was not a main focus of the developed solutions (as discussed in Section 8.3), further work should be performed on these aspects. That also means that the developed solutions should be evaluated on extra aspects (Hassenzahl et al., 2010), such as user experience (UX), emotional reaction of users during execution of tasks with new technologies, etc. Towards that direction, Task Technology Fit (TTF) (Goodhue & Thompson, 1995) and the contemporary TAM-UX (Mazmela et al., 2018) evaluation models shall be considered.

8.5 Final remarks

This final section concludes this thesis with a few final remarks. Section 8.5.1 discusses lessons learnt through the course of the entire research project. Section 8.5.2 delivers a takeaway message.

8.5.1 Lessons learnt

The research presented in this thesis performed within the frames of three European projects (Section 2.2.1). Through the entire process of designing, implementing, testing, deploying and demonstrating MPMS and the complete CPS solutions, and through the collaboration with various project partners with diverse backgrounds (with respect to the process view of K4+1 framework - Figure 81), valuable experiences have been gained that might be useful for the adoption of the solutions. The most important points are highlighted below:

- The distinction between global and local levels is important to separate concepts and keep structured hierarchy levels. Modeling a process from a global perspective (with MPMS) provides a good overview of activities to process owners and production managers. Modeling the physical activities with a more detailed view (e.g., with a state machine local orchestrator component) provides a clear view on how things happen, which is important for the process participants (i.e., human operators) as well. However, the line between these two levels is not always clear. That is mostly obvious on modeling tasks and steps. For instance, a modeler could combine the two consecutive tasks, “Move to wagon with trays” and “Pick-up trays”, of Figure 160, into one, e.g., “Fetch trays”, depending on how much control is desired on the global level and whether the mobile robot (AGV plus mounter robotic arm) should be treated as a single actor. Thus, the granularity of the process models and workflows might be a challenge. However, we believe that it is a matter of agreements between process modelers based on the views and the control they want to provide on each level.

- The process-oriented control that a system like MPMS provides, seems to be a useful approach to fill the gap of missing or insufficient process overview, visibility of production status, resource allocation and orchestration, that many manufacturers face on their production sites. The techniques, though, might be hard for some group of people to grasp and adopt. The use of BPMN as a language to model manufacturing processes, while is gaining a lot of interest in the manufacturing domain, requires experienced process modelers. Moreover, operators without knowledge of the notation might have difficulties to understand the modeled workflows. However, the expressiveness of the notation concerning integration and execution semantics (Ko et al., 2009), together with the implementation solutions we provided within advanced MPMS, make BPMN a good candidate for applications by practitioners in smart manufacturing.
- Safety is cornerstone aspect of Industry 4.0 developments. A CPS shall provide functionality to prevent hazards for humans. For example, the HORSE system includes global and local awareness modules, with response time being an important factor to distinguish their functionalities and responsibilities. Critical events with sub-second response times that occur within a work cell need to be addressed by the local awareness module. The module should track and analyze, in real-time, physical movements within a work cell (e.g., with laser curtains, 3D-space or thermal cameras). When there is any imminent human-robot collision, it has the responsibility and the authority to provide the right instructions to the involved agents for immediate and effective action (for instance, stopping the robot or raising emergency alerts). In case safety breaches can impact a bigger area than a work cell, or the risks are not real-time critical, the captured events need to be propagated to the global situation awareness module. However, the involvement of a lot of components in vertical control of physical actors might impact response times. A task assignment message from MPMS, through the Message Bus, through a robot controller software and finally to a robot can cause some latency, as this was practically experienced in some pilots. This has a negative effect not only on performance and efficiency (by increasing cycle times), but also on safety. Thus, the selection of the communication protocols and powerful and robust infrastructures to host the applications (either on-premise or onsite) are important factors.

8.5.2 Takeaway message

Currently, markets for many product categories are becoming extremely dynamic. The electronics and automotive markets are typical examples. This development implies that manufacturers have to become increasingly flexible in their operations. Customers demand more tailor-made products, with shorter delivery times. Manufacturing processes have become more complex to satisfy this demand and enterprises, especially SMEs, have to be reactive to stay competitive. The Industry 4.0 developments with advanced robotics, AGVs and AR systems, leveraged by the Internet-of-Things, Cyber-Physical Systems and Cloud Computing, promise significant gains in production efficiency, manufacturing flexibility and product customization. The realization in industrial practice, though, of these developments is not an easy task, as it faces many challenges, such as technology heterogeneity, lack of digitization, etc. More importantly, robotic solutions are often employed with a bottom-up approach, following a vertical orientation in their robot control processes. This normally leads to isolated, fragmented developments that add extra complexity in the efforts of achieving horizontal process integration.

The research presented in this thesis aims at tackling manufacturing operations complexity in smart production environments. With the application and extension of the well-established business process management paradigm, knowledge artefacts are provided on how practitioners can put structure in their complex production processes, be responsive to market and operational events that disrupt their systems, and orchestrate their activities performed by heterogenous actors. The theory presented is an enabler of horizontal and vertical process integration in smart manufacturing.

Bibliography

Aalst, W. M. P. (1998). Three Good Reasons for Using a Petri-Net-Based Workflow Management System. In *Information and Process Integration in Enterprises* (pp. 161–182). Springer US. https://doi.org/10.1007/978-1-4615-5499-8_10

Aaltonen, I., & Salmi, T. (2019). Experiences and expectations of collaborative robots in industry and academia: barriers and development needs. *Procedia Manufacturing*, *38*, 1151–1158. https://doi.org/10.1016/J.PROMFG.2020.01.204

Abele, E., Liebeck, T., & Wörn, A. (2006). Measuring Flexibility in Investment Decisions for Manufacturing Systems. *CIRP Annals*, *55*(1), 433–436. https://doi.org/10.1016/S0007-8506(07)60452-1

Abouzid, I., & Saidi, R. (2019). Proposal of BPMN extensions for modelling manufacturing processes. *2019 International Conference on Optimization and Applications, ICOA 2019*. https://doi.org/10.1109/ICOA.2019.8727651

Adam, E. E. (1983). Towards a Typology of Production and Operations Management Systems. *Source: The Academy of Management Review*, *8*(3), 365–375. https://about.jstor.org/terms

Adams, W. C. (2015). Conducting Semi-Structured Interviews. *Handbook of Practical Program Evaluation: Fourth Edition*, 492–505. https://doi.org/10.1002/9781119171386.CH19

Adeyeri, M. K., Mpofu, K., & Adenuga Olukorede, T. (2015). Integration of agent technology into manufacturing enterprise: A review and platform for industry 4.0. *IEOM 2015 - 5th International Conference on Industrial Engineering and Operations Management, Proceeding*. https://doi.org/10.1109/IEOM.2015.7093910

Aguilar-Savén, R. S. (2004). Business process modelling: Review and framework. *International Journal of Production Economics*, *90*(2), 129–149. https://doi.org/10.1016/S0925-5273(03)00102-6

Ahuett-Garza, H., & Kurfess, T. (2018). A brief discussion on the trends of habilitating technologies for Industry 4.0 and Smart manufacturing. *Manufacturing Letters*, *15*, 60–63. https://doi.org/10.1016/j.mfglet.2018.02.011

Aldin, L., & de Cesare, S. (2011). A literature review on business process modelling: New frontiers of reusability. *Enterprise Information Systems*, *5*(3), 359–383. https://doi.org/10.1080/17517575.2011.557443

Alexakos, C., Georgoudakis, M., Kalogeras, A., Charatsis, K., Gialelis, J., & Koubias, S. (2006). A model for the extension of IEC 62264 down to the shop floor layer. *IEEE International Workshop on Factory Communication Systems - Proceedings, WFCS*, 243–246. https://doi.org/10.1109/WFCS.2006.1704162

Andaloussi, A. A., Burattin, A., Slaats, T., Petersen, A. C. M., Hildebrandt, T. T., & Weber, B. (2019). Exploring the Understandability of a Hybrid Process Design Artifact Based on DCR Graphs. *Lecture Notes in Business Information Processing*, *352*, 69–84. https://doi.org/10.1007/978-3-030-20618-5_5

Andree, K., Ihde, S., & Pufahl, L. (2020). Exception handling in the context of fragment-based case management. In *Lecture Notes in Business Information Processing: Vol. 387 LNBIP*. Springer International Publishing. https://doi.org/10.1007/978-3-030-49418-6_2

Angelov, S., Grefen, P., & Greefhorst, D. (2012). A framework for analysis and design of software reference architectures. *Information and Software Technology*, *54*(4), 417–431. https://doi.org/10.1016/J.INFSOF.2011.11.009

Angkiriwang, R., Pujawan, I. N., & Santosa, B. (2014). Managing uncertainty through supply chain flexibility: reactive vs. proactive approaches. *Production and Manufacturing Research*, *2*(1), 50–70. https://doi.org/10.1080/21693277.2014.882804

Araújo, M. B., & Gonçalves, R. F. (2016). Selecting a Notation to Modeling Business Process: A Systematic Literature Review of Technics and Tools. *IFIP Advances in Information and Communication Technology*, *488*, 198–205. https://doi.org/10.1007/978-3-319-51133-7_24

Arnaudov, V. (2018a). *Final Version of HORSE Cross-Domain Messaging. Deliverable D3.11. HORSE Consortium*. (confidential deliverable)

Arnaudov, V. (2018b). *User Handbook. Deliverable D4.5. HORSE Consortium*. http://horse-project.eu/sites/default/files/publications/HORSE_D4.5 User Handbook v1.00.pdf
Arnaudov, V. (2019). *Test Report. Deliverable D4.4. HORSE Consortium*. (confidential deliverable)
Årzén, K.-E. (1996). Grafchart: A Graphical Language for Sequential Supervisory Control Applications. *IFAC Proceedings Volumes*, *29*(1), 4831–4836. https://doi.org/10.1016/S1474-6670(17)58445-1
Aspridou, M. (2017). *Extending BPMN for modeling manufacturing processes* [Master thesis, Eindhoven University of Technology]. https://research.tue.nl/en/studentTheses/extending-bpmn-for-modeling-manufacturing-processes
Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A survey. *Computer Networks*, *54*(15), 2787–2805. https://doi.org/10.1016/j.comnet.2010.05.010
Avvaru, V. S., Bruno, G., Chiabert, P., & Traini, E. (2020). Integration of PLM, MES and ERP Systems to Optimize the Engineering, Production and Business. *IFIP Advances in Information and Communication Technology*, *594*, 70–82. https://doi.org/10.1007/978-3-030-62807-9_7
Babar, M. A., & Gorton, I. (2004). Comparison of scenario-based software architecture evaluation methods. *Proceedings - Asia-Pacific Software Engineering Conference, APSEC*, 600–607. https://doi.org/10.1109/APSEC.2004.38
Babiceanu, R. F., & Seker, R. (2016). Big Data and virtualization for manufacturing cyber-physical systems: A survey of the current status and future outlook. *Computers in Industry*, *81*, 128–137. https://doi.org/10.1016/J.COMPIND.2016.02.004
Bader, S. R., Maleshkova, M., & Lohmann, S. (2019). Structuring Reference Architectures for the Industrial Internet of Things. *Future Internet*, *11*(7), 151. https://doi.org/10.3390/fi11070151
Barbosa, J., Leitão, P., Adam, E., & Trentesaux, D. (2015). Dynamic self-organization in holonic multi-agent manufacturing systems: The ADACOR evolution. *Computers in Industry*, *66*, 99–111. https://doi.org/10.1016/J.COMPIND.2014.10.011
Bauer, M., Boussard, M., Lucent, A., Bui, N., & Carrez, F. (2013). *Internet of Things – Architecture IoT-A Deliverable D1 . 5 – Final architectural Internet of Things – Architecture IoT-A* (Issue July). https://www.researchgate.net/publication/272814818_Internet_of_Things_-_Architecture_IoT-A_Deliverable_D15_-_Final_architectural_reference_model_for_the_IoT_v30
Bauer, W., Bender, M., Braun, M., Rally, P., & Scholtz, O. (2016). *Lightweight robots in manual assembly – best to start simply! Examining companies' initial experiences with lightweight robots*. https://www.researchgate.net/publication/327744724_Lightweight_robots_in_manual_assembly_-_best_to_start_simply_Examining_companies%27_initial_experiences_with_lightweight_robots
Baumgraß, A., Botezatu, M., di Ciccio, C., Dijkman, R., Grefen, P., Hewelt, M., Mendling, J., Meyer, A., Pourmirza, S., & Völzer, H. (2016). Towards a Methodology for the Engineering of Event-Driven Process Applications. *Lecture Notes in Business Information Processing*, *256*, 501–514. https://doi.org/10.1007/978-3-319-42887-1_40
Baumgraß, A., Dijkman, R., Grefen, P., Pourmirza, S., Völzer, H., & Weske, M. (2015). A software architecture for transportation planning and monitoring in a collaborative network. *IFIP Advances in Information and Communication Technology*, *463*, 277–284. https://doi.org/10.1007/978-3-319-24141-8_25
Bejarano, R., Ferrer, B. R., Mohammed, W. M., & Martinez Lastra, J. L. (2019). Implementing a human-robot collaborative assembly workstation. *IEEE International Conference on Industrial Informatics (INDIN)*, *2019-July*, 557–564. https://doi.org/10.1109/INDIN41052.2019.8972158
Block, C., Lins, D., & Kuhlenkötter, B. (2018). Approach for a simulation-based and event-driven production planning and control in decentralized manufacturing execution systems. *Procedia CIRP*, *72*, 1351–1356. https://doi.org/10.1016/J.PROCIR.2018.03.204
Boiko, O., Shendryk, V., Shendryk, S., & Boiko, A. (2020). Mes/erp integration aspects of the manufacturing automation. *Lecture Notes in Mechanical Engineering*, 15–24. https://doi.org/10.1007/978-3-030-40724-7_2

Böllhoff, J., Metternich, J., Frick, N., & Kruczek, M. (2016). Evaluation of the Human Error Probability in Cellular Manufacturing. *Procedia CIRP*, *55*, 218–223. https://doi.org/10.1016/J.PROCIR.2016.07.080

Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012). Fog computing and its role in the internet of things. *MCC'12 - Proceedings of the 1st ACM Mobile Cloud Computing Workshop*, 13–15. https://doi.org/10.1145/2342509.2342513

Booch, G., Rumbaugh, J., & Jacobson. I. (1999). The Unified Modeling Language User Guide. Reading-Massachusetts. *Addison Wesley.* http://www.awl.com/cseng/

Boote, D. N., & Beile, P. (2005). Scholars Before Researchers: On the Centrality of the Dissertation Literature Review in Research Preparation. *Educational Researcher*, *34*(6), 3–15. https://doi.org/10.3102/0013189X034006003

Braglia, M., Carmignani, G., & Zammori, F. (2006). A new value stream mapping approach for complex production systems. *International Journal of Production Research*, *44*(18–19), 3929–3952. https://doi.org/10.1080/00207540600690545

Brahe, S. (2007). BPM on top of SOA: Experiences from the financial industry. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *4714 LNCS*, 96–111. https://doi.org/10.1007/978-3-540-75183-0_8

Braun, R., Schlieter, H., Burwitz, M., & Esswein, W. (2014). BPMN4CP: Design and implementation of a BPMN extension for clinical pathways. *Proceedings - 2014 IEEE International Conference on Bioinformatics and Biomedicine, IEEE BIBM 2014*, 9–16. https://doi.org/10.1109/BIBM.2014.6999261

Brettel, M., Heinemann, F., Engelen, A., & Neubauer, S. (2011). Cross-Functional Integration of R&D, Marketing, and Manufacturing in Radical and Incremental Product Innovations and Its Effects on Project Effectiveness and Efficiency. In *Journal of Product Innovation Management* (Vol. 28, Issue 2). John Wiley & Sons, Ltd. https://doi.org/10.1111/J.1540-5885.2011.00795.X

Brocke, J. vom, & Rosemann, M. (2010). Handbook on Business Process Management 1. In J. vom Brocke & M. Rosemann (Eds.), *International Handbooks on Information Systems Series*. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-00416-2

Brouns, N. (2019). *On the road towards Smart Manufacturing A Framework to support the development of Smart Manufacturing*. https://research.tue.nl/en/studentTheses/on-the-road-towards-smart-manufacturing

Brox, J. A., Fader, C., Broxy, J. A., & Fadery, C. (2010). The set of just-in-time management strategies: An assessment of their impact on plant-level productivity and input-factor substitutability using variable cost function estimates. *Https://Doi.Org/10.1080/00207540210137657*, *40*(12), 2705–2720. https://doi.org/10.1080/00207540210137657

Bruccoleri, M., Amico, M., Perrone, G., Bruccoleriy, M., Amicoy, M., & Perronez, G. (2003). Distributed intelligent control of exceptions in reconfigurable manufacturing systems. *International Journal of Production Research*, *41*(7), 1393–1412. https://doi.org/10.1080/1352816031000075170

Bruccoleri, M., Renna, P., & Perrone, G. (2007). Reconfiguration: a key to handle exceptions and performance deteriorations in manufacturing operations. *Http://Dx.Doi.Org/10.1080/00207540500140773*, *43*(19), 4125–4145. https://doi.org/10.1080/00207540500140773

Bruno, G., & Antonelli, D. (2018). Dynamic task classification and assignment for the management of human-robot collaborative teams in workcells. *International Journal of Advanced Manufacturing Technology*, *98*(9–12), 2415–2427. https://doi.org/10.1007/s00170-018-2400-4

Brunoe, T. D., & Nielsen, K. (2016). Complexity Management in Mass Customization SMEs. *Procedia CIRP*, *51*, 38–43. https://doi.org/10.1016/j.procir.2016.05.099

Bucchiarone, A., Cabot, J., Paige, R. F., & Pierantonio, A. (2020). Grand challenges in model-driven engineering: an analysis of the state of the research. *Software and Systems Modeling*, *19*(1), 5–13. https://doi.org/10.1007/S10270-019-00773-6/FIGURES/1

Cachon, G., & Terwiesch, C. (2009). Matching supply with demand : an introduction to operations management. In *Matching supply with demand*. McGraw-Hill/Irwin,.

Cadavid, J., Alférez, M., Gérard, S., & Tessier, P. (2015). Conceiving the model-driven smart factory. *Proceedings of the 2015 International Conference on Software and System Process*, *24-26-Augu*, 72–76. https://doi.org/10.1145/2785592.2785602
Caiza, G., Saeteros, M., Oñate, W., & Garcia, M. v. (2020). Fog computing at industrial level, architecture, latency, energy, and security: A review. *Heliyon*, *6*(4), e03706. https://doi.org/10.1016/J.HELIYON.2020.E03706
Calvary, G., Dery-Pinna, A.-M., Occello, A., Renevier-Gonin, P., & Riveill, M. (2014). At the Cross-Roads between Human-Computer Interaction and Model Driven Engineering. *ARPN Journal of Systems and Software*, *4*(3). http://www.scientific-journals.org
Calvo Olivares, R. D., Rivera, S. S., & Núñez Mc Leod, J. E. (2018). A novel qualitative prospective methodology to assess human error during accident sequences. *Safety Science*, *103*, 137–152. https://doi.org/10.1016/J.SSCI.2017.10.023
Carvalho, A., O' Mahony, N., Krpalkova, L., Campbell, S., Walsh, J., & Doody, P. (2019). Edge Computing Applied to Industrial Machines. *Procedia Manufacturing*, *38*, 178–185. https://doi.org/10.1016/J.PROMFG.2020.01.024
Carvalho, N., Chaim, O., Cazarini, E., & Gerolamo, M. (2018). Manufacturing in the fourth industrial revolution: A positive prospect in Sustainable Manufacturing. *Procedia Manufacturing*, *21*, 671–678. https://doi.org/10.1016/j.promfg.2018.02.170
Castro, S., & Teixeira, L. (2021). Industry 4.0 and business process management: An exploratory study on the bilateral effects. *Proceedings of the International Conference on Industrial Engineering and Operations Management*, 4840–4847.
Chen †, D. (2005). Enterprise-control system integration—an international standard. *International Journal of Production Research*, *43*(20), 4335–4357. https://doi.org/10.1080/00207540500142399
Chinosi, M., & Trombetta, A. (2012). BPMN: An introduction to the standard. *Computer Standards & Interfaces*, *34*(1), 124–134. https://doi.org/10.1016/J.CSI.2011.06.002
Choi, B. K., & Kim, B. H. (2010). MES (manufacturing execution system) architecture for FMS compatible to ERP (enterprise planning system). *Https://Doi.Org/10.1080/09511920110059106*, *15*(3), 274–284. https://doi.org/10.1080/09511920110059106
Choi, T. M., Wang, M., & Yue, X. (2016). Emerging production optimization issues in supply chain systems. *Annals of Operations Research*, *240*(2), 381–393. https://doi.org/10.1007/s10479-015-1948-8
Christopher, M., & Ryals, L. J. (2014). The supply chain becomes the demand chain. In *Journal of Business Logistics* (Vol. 35, Issue 1, pp. 29–35). Wiley-Blackwell. https://doi.org/10.1111/jbl.12037
Chryssolouris, G., Efthymiou, K., Papakostas, N., Mourtzis, D., & Pagoropoulos, A. (2013). Flexibility and complexity: Is it a trade-off? *International Journal of Production Research*, *51*(23–24), 6788–6802. https://doi.org/10.1080/00207543.2012.761362
Chung, P. W. H., Cheung, L., Stader, J., Jarvis, P., Moore, J., & Macintosh, A. (2003). Knowledge-based process management—an approach to handling adaptive workflow. *Knowledge-Based Systems*, *16*(3), 149–160. https://doi.org/10.1016/S0950-7051(02)00080-1
Cimino, C., Negri, E., & Fumagalli, L. (2019). Review of digital twin applications in manufacturing. *Computers in Industry*, *113*, undefined-undefined. https://doi.org/10.1016/J.COMPIND.2019.103130
Ciurana, J., Garcia-Romeu, M. L., Ferrer, I., & Casadesús, M. (2008). A model for integrating process planning and production planning and control in machining processes. *Robotics and Computer-Integrated Manufacturing*, *24*(4), 532–544. https://doi.org/10.1016/J.RCIM.2007.07.013
Cohn, D., & Hull, R. (2009). Business artifacts: A data-centric approach to modeling business operations and processes. *IEEE Data Eng. Bull*, *32*(3), 3–9. http://scholar.google.com/scholar?hl=en%257B&%257DbtnG=Search%257B&%257Dq=intitle:Business+Artifacts+:+A+Data-centric+Approach+to+Modeling+Business+Operations+and+Processes%257B#%257D0

Colombo, A. W., Schoop, R., & Neubert, R. (2006). An agent-based intelligent control platform for industrial holonic manufacturing systems. *IEEE Transactions on Industrial Electronics*, *53*(1), 322–337. https://doi.org/10.1109/TIE.2005.862210

da Xu, L., Xu, E. L., & Li, L. (2018). Industry 4.0: state of the art and future trends. *International Journal of Production Research*, *56*(8), 2941–2962. https://doi.org/10.1080/00207543.2018.1444806

Dalenogare, L. S., Benitez, G. B., Ayala, N. F., & Frank, A. G. (2018). The expected contribution of Industry 4.0 technologies for industrial performance. *International Journal of Production Economics*, *204*, 383–394. https://doi.org/10.1016/j.ijpe.2018.08.019

Dalmarco, G., Ramalho, F. R., Barros, A. C., & Soares, A. L. (2019). Providing industry 4.0 technologies: The case of a production technology cluster. *Journal of High Technology Management Research*, *30*(2), 100355. https://doi.org/10.1016/j.hitech.2019.100355

Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly: Management Information Systems*, *13*(3), 319–339. https://doi.org/10.2307/249008

Davis, J., Edgar, T., Porter, J., Bernaden, J., & Sarli, M. (2012). Smart manufacturing, manufacturing intelligence and demand-dynamic performance. *Computers and Chemical Engineering*, *47*, 145–156. https://doi.org/10.1016/j.compchemeng.2012.06.037

de Groote, X. (1989). *Inventory Theory: A Road Map. Teaching Note*.

de Snoo, C., van Wezel, W., Wortmann, J. C., & Gaalman, G. J. C. (2010). Coordination activities of human planners during rescheduling: case analysis and event handling procedure. *Http://Dx.Doi.Org/10.1080/00207541003639626*, *49*(7), 2101–2122. https://doi.org/10.1080/00207541003639626

de Ugarte, B., Artiba, A., & Pellerin, R. (2009). Manufacturing execution system – a literature review. *Production Planning & Control*, *20*(6), 525–539. https://doi.org/10.1080/09537280902938613

Decker, G., & Barros, A. (2008). Interaction modeling using BPMN. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *4928 LNCS*, 208–219. https://doi.org/10.1007/978-3-540-78238-4_22

Defense Acquisition University DAU. (2021). *Integrated Product Support (IPS) Elements Guidebook*.

DeMarco, T. (1979). Structure Analysis and System Specification. *Pioneers and Their Contributions to Software Engineering*, 255–288. https://doi.org/10.1007/978-3-642-48354-7_9

Devos, J., van Landeghem, H., & Deschoolmeester, D. (2014). *Information Systems for Small and Medium-sized Enterprises* (J. Devos, H. van Landeghem, & D. Deschoolmeester, Eds.). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-38244-4

Dey, A. K., Hamid, R., Beckmann, C., Li, I., & Hsu, D. (2004). a CAPpella: Programming by Demonstration of Context-Aware Applications. *Proceedings of the 2004 Conference on Human Factors in Computing Systems - CHI '04*.

Dillmann, R., & Friedrich, H. (1996). Programming by demonstration: A machine learning approach to support skill acquision for robots. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *1138*, 87–108. https://doi.org/10.1007/3-540-61732-9_52

DIN/DKE. (2016). *German Standardization Roadmap*. https://www.din.de/resource/blob/65354/1bed7e8d800cd4712d7d1786584a7a3a/roadmap-i4-0-e-data.pdf

DKE. (2015). *National Intelligent Manufacturing Standard System Construction Guidelines*. https://www.dke.de/resource/blob/929020/7080b1667308545c088901b39a111756/manufacturing-guidelines-data.pdf

Dobrica, L., & Niemelá, E. (2002). A survey on software architecture analysis methods. *IEEE Transactions on Software Engineering*, *28*(7), 638–653. https://doi.org/10.1109/TSE.2002.1019479

Dumas, M., la Rosa, M., Mendling, J., & Reijers, H. A. (2018). *Fundamentals of Business Process Management*. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-56509-4

Duray, R., Ward, P. T., Milligan, G. W., & Berry, W. L. (2000). Approaches to mass customization: configurations and empirical validation. *Journal of Operations Management*, *18*, 605–625. www.elsevier.comrlocaterdsw
Eder, J., & Liebhart, W. (1995). The Workflow Activity Model (WAMO). *3rd International Conference on Cooperative Information Systems (CoopIs '95)*, 87–98.
el Maraghy, H. A. (2006). Flexible and reconfigurable manufacturing systems paradigms. *International Journal of Flexible Manufacturing Systems 2006 17:4*, *17*(4), 261–276. https://doi.org/10.1007/S10696-006-9028-7
el Zaatari, S., Marei, M., Li, W., & Usman, Z. (2019). Cobot programming for collaborative industrial tasks: An overview. *Robotics and Autonomous Systems*, *116*, 162–180. https://doi.org/10.1016/j.robot.2019.03.003
ElMaraghy, H. A. (2009). Changing and Evolving Products and Systems – Models and Enablers. *Changeable and Reconfigurable Manufacturing Systems*, 25–45. https://doi.org/10.1007/978-1-84882-067-8_2
ElMaraghy, H., Schuh, G., Elmaraghy, W., Piller, F., Schönsleben, P., Tseng, M., & Bernard, A. (2013). Product variety management. *CIRP Annals - Manufacturing Technology*, *62*(2), 629–652. https://doi.org/10.1016/j.cirp.2013.05.007
Elmaraghy, W., Elmaraghy, H., Tomiyama, T., & Monostori, L. (2012). Complexity in engineering design and manufacturing. *CIRP Annals*, *61*(2), 793–814. https://doi.org/10.1016/J.CIRP.2012.05.001
Engels, G., Strothmann, T., & Teetz, A. (2018). Adapt Cases 4 BPM - A Modeling Framework for Process Flexibility in IIoT. *Proceedings - IEEE International Enterprise Distributed Object Computing Workshop, EDOCW*, *2018-Octob*, 59–68. https://doi.org/10.1109/EDOCW.2018.00020
Entringer, T. C., Ferreira, A. da S., & Nascimento, D. C. de O. (2021). Comparative analysis of the main business process modeling methods: a bibliometric study. *Gestão & Produção*, *28*(2), 2021. https://doi.org/10.1590/1806-9649-2020v28e5211
Erasmus, J. (2019). *The application of business process management in smart manufacturing - Towards dynamic allocation of manufacturing resources* [Technical University of Eindhoven]. https://pure.tue.nl/ws/portalfiles/portal/138259266/20191119_Erasmus.pdf
Erasmus, J., Grefen, P., Vanderfeesten, I., & Traganos, K. (2018). Smart Hybrid Manufacturing Control Using Cloud Computing and the Internet-of-Things. *Machines 2018, Vol. 6, Page 62*, *6*(4), 62. https://doi.org/10.3390/MACHINES6040062
Erasmus, J., Vanderfeesten, I., Traganos, K., & Grefen, P. (2018). The Case for Unified Process Management in Smart Manufacturing. *2018 IEEE 22nd International Enterprise Distributed Object Computing Conference (EDOC)*, 218–227. https://doi.org/10.1109/EDOC.2018.00035
Erasmus, J., Vanderfeesten, I., Traganos, K., & Grefen, P. (2020). Using business process models for the specification of manufacturing operations. *Computers in Industry*, *123*, 103297. https://doi.org/10.1016/j.compind.2020.103297
Erasmus, J., Vanderfeesten, I., Traganos, K., Jie-A-Looi, X., Kleingeld, A., & Grefen, P. (2018). A Method to Enable Ability-Based Human Resource Allocation in Business Process Management Systems. *Lecture Notes in Business Information Processing*, *335*, 37–52. https://doi.org/10.1007/978-3-030-02302-7_3
Erasmus, J., Vanderfeesten, I., Traganos, K., Keulen, R., & Grefen, P. (2020). The HORSE Project: The Application of Business Process Management for Flexibility in Smart Manufacturing. *Applied Sciences 2020, Vol. 10, Page 4145*, *10*(12), 4145. https://doi.org/10.3390/APP10124145
Estruch, A., & Heredia Álvaro, J. A. (2012). Event-driven manufacturing process management approach. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *7481 LNCS*, 120–133. https://doi.org/10.1007/978-3-642-32885-5_9
European Commission. (2021). *Annual Report on European SMEs 2020/2021: Digitalisation of SMEs* (Issue July). https://doi.org/10.2826/515420

European Dynamics. (2018). *First system deployed at factories sites. Deliverable D5.2. HORSE Project*. http://horse-project.eu/sites/default/files/publications/HORSE_D5.2-v1.00.pdf
Fahland, D. (2019). Describing Behavior of Processes with Many-to-Many Interactions. In S. Donatelli & S. Haar (Eds.), *Application and Theory of Petri Nets and Concurrency. PETRI NETS 2019: Vol. 11522 LNCS* (pp. 3–24). Springer Verlag. https://doi.org/10.1007/978-3-030-21571-2_1/FIGURES/9
Fahland, D., de Leoni, M., van Dongen, B. F., & van der Aalst, W. M. P. (2011). Many-to-Many: Some Observations on Interactions in Artifact Choreographies. In D. Eichhorn, A. Koschmider, & H. Zhang (Eds.), *Proceedings of the 3rd Central-European Workshop on Services and their Composition, ZEUS 2011* (pp. 9–15). CEUR Workshop Proceedings.
Fahland, D., Lübke, D., Mendling, J., Reijers, H., Weber, B., Weidlich, M., & Zugal, S. (2009). Declarative versus Imperative Process Modeling Languages: The Issue of Understandability. *Lecture Notes in Business Information Processing*, *29 LNBIP*, 353–366. https://doi.org/10.1007/978-3-642-01862-6_29
Fang, B., Guo, D., Sun, F., Liu, H., & Wu, Y. (2015). A robotic hand-arm teleoperation system using human arm/hand with a novel data glove. *2015 IEEE International Conference on Robotics and Biomimetics, IEEE-ROBIO 2015*, 2483–2488. https://doi.org/10.1109/ROBIO.2015.7419712
Farooqui, A., Bergagard, P., Falkman, P., & Fabian, M. (2016). Error handling within highly automated automotive industry: Current practice and research needs. *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*, *2016-Novem*. https://doi.org/10.1109/ETFA.2016.7733628
Fleischmann, A. (2012). From subject-phase model based process specifications to an executable workflow. *Lecture Notes in Business Information Processing*, *104 LNBIP*, 77–90. https://doi.org/10.1007/978-3-642-29133-3_6
Fleischmann, A., Schmidt, W., & Stary, C. (2015). Subject-Oriented Business Process Management. *Handbook on Business Process Management 2: Strategic Alignment, Governance, People and Culture, Second Edition*, 601–621. https://doi.org/10.1007/978-3-642-45103-4_25
Ford, H. (2019). *Today and Tomorrow: Commemorative Edition of Ford's 1926 Classic.* Routledge.
Fraga-Lamas, P., Fernández-Caramés, T. M., Blanco-Novoa, Ó., & Vilar-Montesinos, M. A. (2018). A Review on Industrial Augmented Reality Systems for the Industry 4.0 Shipyard. *IEEE Access*, *6*, 13358–13375. https://doi.org/10.1109/ACCESS.2018.2808326
Fragapane, G., Ivanov, D., Peron, M., Sgarbossa, F., & Strandhagen, J. O. (2020). Increasing flexibility and productivity in Industry 4.0 production networks with autonomous mobile robots and smart intralogistics. *Annals of Operations Research*. https://doi.org/10.1007/s10479-020-03526-7
Fraile, Sanchis, Poler, & Ortiz. (2019). Reference Models for Digital Manufacturing Platforms. *Applied Sciences*, *9*(20), 4433. https://doi.org/10.3390/app9204433
Franciosi, C., di Pasquale, V., Iannone, R., & Miranda, S. (2019). A taxonomy of performance shaping factors for human reliability analysis in industrial maintenance. *Journal of Industrial Engineering and Management*, *12*(1), 115–132. https://doi.org/10.3926/jiem.2702
Frank, A. G., Dalenogare, L. S., & Ayala, N. F. (2019). Industry 4.0 technologies: Implementation patterns in manufacturing companies. *International Journal of Production Economics*, *210*, 15–26. https://doi.org/10.1016/J.IJPE.2019.01.004
Froschauer, R., & Lindorfer, R. (2019). Workflow-based programming of human-robot interaction for collaborative assembly stations. *ARW & OAGM Workshop 2019*. https://doi.org/10.3217/978-3-85125-663-5-14
Gao, M., & Jiang, J. (2009). Towards content-oriented integration for process-centric knowledge management systems. *Proceedings - 2009 Asia-Pacific Conference on Information Processing, APCIP 2009*, *1*, 529–532. https://doi.org/10.1109/APCIP.2009.136
Garcia, A., Sant'Anna, C., Chavez, C., da Silva, V. T., de Lucena, C. J. P., & von Staa, A. (2004). Separation of Concerns in Multi-agent Systems: An Empirical Study. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *2940*, 49–72. https://doi.org/10.1007/978-3-540-24625-1_4

García-Domínguez, A., Marcos, M., Medina, I., & Marcos, M. (2012). *A comparison of BPMN 2.0 with other notations for manufacturing processes*. 601. https://doi.org/10.1063/1.4707613
Gažová, A., Papulová, Z., & Smolka, D. (2022). Effect of Business Process Management on Level of Automation and Technologies Connected to Industry 4.0. *Procedia Computer Science*, *200*, 1498–1507. https://doi.org/10.1016/J.PROCS.2022.01.351
Gerber, T., Theorin, A., Johnsson, C., Gerber, T., Theorin, A., Johnsson, · C, & Johnsson, C. (2014). Towards a seamless integration between process modeling descriptions at business and production levels: work in progress. *J Intell Manuf*, *25*, 1089–1099. https://doi.org/10.1007/s10845-013-0754-x
Glykas, M. (2004). Workflow and process management in printing and publishing firms. *International Journal of Information Management*. https://doi.org/10.1016/j.ijinfomgt.2004.08.003
Goodhue, D. L., & Thompson, R. L. (1995). Task-Technology Fit and Individual Performance Task. *MIS Quarterly*, *19*(2), 213–236. https://www.jstor.org/stable/249689
Gorecky, D., Schmitt, M., Loskyll, M., & Zuhlke, D. (2014). Human-machine-interaction in the industry 4.0 era. *2014 12th IEEE International Conference on Industrial Informatics (INDIN)*, 289–294. https://doi.org/10.1109/INDIN.2014.6945523
Gottumukkala, R. N., & Sun, T. (2005). Modeling and Assessment of Production Printing Workflows Using Petri Nets. *Lecture Notes in Computer Science*, *3649*, 319–333. https://doi.org/10.1007/11538394_21
Graja, I., Kallel, S., Guermouche, N., & Kacem, A. H. (2016). BPMN4CPS: A BPMN Extension for Modeling Cyber-Physical Systems. *2016 IEEE 25th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, 152–157. https://doi.org/10.1109/WETICE.2016.41
Grauer, M., Karadgi, S., Metz, D., & Schäfer, W. (2010). Online Monitoring and Control of Enterprise Processes in Manufacturing Based on an Event-Driven Architecture. In M. zur Muehlen & J. Su (Eds.), *Business Process Management Workshops. BPM 2010: Vol. 66 LNBIP* (pp. 671–682). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-20511-8_61
Grefen, P. (2013). Networked Business Process Management. *International Journal of IT/Business Alignment and Governance*, *4*(2), 54–82. https://doi.org/10.4018/ijitbag.2013070104
Grefen, P. (2015). *Service-Dominant Business Engineering with BASE/X*. CreateSpace Independent Publishing Platform.
Grefen, P. (2016). *Business Information System Architecture (Fall 2016 edition)* (Fall editi).
Grefen, P., & Boultadakis, G. (2021). *Designing an Integrated System for Smart Industry: The Development of the HORSE Architecture*. Independently Published. https://research.tue.nl/nl/publications/designing-an-integrated-system-for-smart-industry-the-development
Grefen, P., Eshuis, R., Mehandjiev, N., Kouvas, G., & Weichhart, G. (2009). Internet-based support for process-oriented instant virtual enterprises. *IEEE Internet Computing*, *13*(6), 65–73. https://doi.org/10.1109/MIC.2009.96
Grefen, P., Mehandjiev, N., Kouvas, G., Weichhart, G., & Eshuis, R. (2009). Dynamic business network process management in instant virtual enterprises. *Computers in Industry*, *60*(2), 86–103. https://doi.org/10.1016/j.compind.2008.06.006
Grefen, P., & Remmerts De Vries, R. N. (1998). A reference architecture for workflow management systems. *Data and Knowledge Engineering*, *27*(1), 31–57. https://doi.org/10.1016/S0169-023X(97)00057-8
Grefen, P., Vanderfeesten, I., Traganos, K., Domagala-Schmidt, Z., & Vleuten, J. van der. (2022). Advancing Smart Manufacturing in Europe: Experiences from Two Decades of Research and Innovation Projects. *Machines 2022, Vol. 10, Page 45*, *10*(1), 45. https://doi.org/10.3390/MACHINES10010045
Grefen, P., Wilbik, A., Kuitems, F., & Blanken, M. (2021). Outcome-Based Business Design in IoT-Enabled Digital Supply Chain Transformation. *2021 IEEE International Conference on*

Internet of Things and Intelligence Systems (IoTaIS), 28–34. https://doi.org/10.1109/IOTAIS53735.2021.9628770
Gregor, M., Hercko, J., & Grznar, P. (2015). The factory of the future production system research. *2015 21st International Conference on Automation and Computing: Automation, Computing and Manufacturing for New Economic Growth, ICAC 2015*. https://doi.org/10.1109/ICONAC.2015.7313998
Gregor, S. (2006). The Nature of Theory in Information Systems. *Source: MIS Quarterly*, *30*(3), 611–642.
Gregor, S., & Hevner, A. R. (2013). Positioning and Presenting Design Science Research for Maximum Impact. In *Source: MIS Quarterly* (Vol. 37, Issue 2).
Groover, M. (2010). Fundementals of Modern Manufacturing Materials,Processes and Systems. *John Wiley & Sons*, 493.
Gualtieri, L., Palomba, I., Wehrle, E. J., & Vidoni, R. (2020). The Opportunities and Challenges of SME Manufacturing Automation: Safety and Ergonomics in Human–Robot Collaboration. *Industry 4.0 for SMEs: Challenges, Opportunities and Requirements*, 105–144. https://doi.org/10.1007/978-3-030-25425-4_4
Gupta, Y. P., & Goyal, S. (1989). Flexibility of manufacturing systems: Concepts and measurements. *European Journal of Operational Research*, *43*, 119–135.
Gupta, Y. P., & Somers, T. M. (1992). The measurement of manufacturing flexibility. *European Journal of Operational Research*, *60*, 166–182.
Haarmann, S., Podlesny, N. J., Hewelt, M., Meyer, A., & Weske, M. (2015). Production case management: A prototypical process engine to execute flexible business processes. *CEUR Workshop Proceedings*, *1418*(January), 110–114.
Hankel, M., & Rexroth, B. (2015). *The Reference Architectural Model Industrie 4.0 (RAMI 4.0)*. https://przemysl-40.pl/wp-content/uploads/2010-The-Reference-Architectural-Model-Industrie-40.pdf
Hans, R., Abels, S., Zöller, S., Miede, A., & Steinmetz, R. (2013). Enabling collaboration in virtual manufacturing enterprises with cloud computing. *19th Americas Conference on Information Systems, AMCIS 2013 - Hyperconnected World: Anything, Anywhere, Anytime*, *2*, 1341–1350.
Harmon, P. (2010). The Scope and Evolution of Business Process Management. In *Handbook on Business Process Management 1* (pp. 37–81). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-00416-2_3
Hartmann, L., Meudt, T., Seifermann, S., & Metternich, J. (2018). Value stream method 4.0: Holistic method to analyse and design value streams in the digital age. *Procedia CIRP*, *78*, 249–254. https://doi.org/10.1016/J.PROCIR.2018.08.309
Hassenzahl, M., Diefenbach, S., & Göritz, A. (2010). Needs, affect, and interactive products - Facets of user experience. *Interacting with Computers*, *22*(5), 353–362. https://doi.org/10.1016/J.INTCOM.2010.04.002
Hausman, W. H., Montgomery, D. B., & Roth, A. v. (2002). Why should marketing and manufacturing work together? *Journal of Operations Management*, *20*(3), 241–257. https://doi.org/10.1016/S0272-6963(02)00010-4
Hayes, R. H., & Wheelwright, S. C. (1984). *Restoring our competitive edge : competing through manufacturing* (Vol. 10). John Wiley & Sons Incorporated.
Hermann, A., Drews, F., Bauer, J., Klemm, S., Roennau, A., & Dillmann, R. (2014). Unified GPU voxel collision detection for mobile manipulation planning. *IEEE International Conference on Intelligent Robots and Systems*, 4154–4160. https://doi.org/10.1109/IROS.2014.6943148
Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. In *Source: MIS Quarterly* (Vol. 28, Issue 1). https://www.jstor.org/stable/25148625
Heyer, C. (2010). Human-robot interaction and future industrial robotics applications. *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings*, 4749–4754. https://doi.org/10.1109/IROS.2010.5651294
Hildebrandt, T., Mukkamala, R. R., & Slaats, T. (2012). Nested dynamic condition response graphs. In F. Arbab & M. Sirjani (Eds.), *Fundamentals of Software Engineering: Vol. 7141 LNCS* (pp. 343–350). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-29320-7_23

Hofmann, E., & Rüsch, M. (2017). Industry 4.0 and the current status as well as future prospects on logistics. *Computers in Industry*, *89*, 23–34. https://doi.org/10.1016/j.compind.2017.04.002
Holdren, J. P., Power, T., Tassey, G., Ratcliff, A., & Christodoulou, L. (2012). *A national strategic plan for advanced manufacturing*. https://obamawhitehouse.archives.gov/sites/default/files/microsites/ostp/iam_advancedmanufacturing_strategicplan_2012.pdf
Hollingsworth, D. (1995). *Workflow Management Coalition: The Workflow Reference Model. Document Number TC00-1003, Document status - Issue 1.1*. https://doi.org/10.1.1.198.5206
Hon, K. K. B. (2005). Performance and Evaluation of Manufacturing Systems. *CIRP Annals*, *54*(2), 139–154. https://doi.org/10.1016/S0007-8506(07)60023-7
Horváth, D., & Szabó, R. Z. (2019). Driving forces and barriers of Industry 4.0: Do multinational and small and medium-sized companies have equal opportunities? *Technological Forecasting and Social Change*, *146*, 119–132. https://doi.org/10.1016/J.TECHFORE.2019.05.021/DRIVING_FORCES_AND_BARRIERS_OF_INDUSTRY_4_0_DO_MULTINATIONAL_AND_SMALL_AND_MEDIUM_SIZED_COMPANIES_HAVE_EQUAL_OPPORTUNITIES_.PDF
Hu, S. J., Ko, J., Weyand, L., Elmaraghy, H. A., Lien, T. K., Koren, Y., Bley, H., Chryssolouris, G., Nasr, N., & Shpitalni, M. (2011). Assembly system design and operations for product variety. *CIRP Annals - Manufacturing Technology*, *60*(2), 715–733. https://doi.org/10.1016/j.cirp.2011.05.004
Hu, S. J., Zhu, X., Wang, H., & Koren, Y. (2008). Product variety and manufacturing complexity in assembly systems and supply chains. *CIRP Annals*, *57*(1), 45–48. https://doi.org/10.1016/J.CIRP.2008.03.138
Huang, Z., Kim, J., Sadri, A., Dowey, S., & Dargusch, M. S. (2019). Industry 4.0: Development of a multi-agent system for dynamic value stream mapping in SMEs. *Journal of Manufacturing Systems*, *52*, 1–12. https://doi.org/10.1016/J.JMSY.2019.05.001
Hull, F. M., & Collins, P. D. (1987). Research Notes. High-Technology Batch Production Systems: Woodward'S Missing Type. *Academy of Management Journal*, *30*(4), 786–797. https://doi.org/10.2307/256160
Hull, R., Damaggio, E., Fournier, F., Gupta, M., Heath, F., Hobson, S., Linehan, M., Maradugu, S., Nigam, A., Sukaviriya, P., & Vaculin, R. (2011). Introducing the guard-stage-milestone approach for specifying business entity lifecycles. In M. Bravetti & T. Bultan (Eds.), *Web Services and Formal Methods: Vol. 6551 LNCS* (pp. 1–24). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-19589-1_1
IEC. (1997). *IEC 61512-1, Batch Control, Part 1: Models and Terminology*. https://standards.globalspec.com/std/185020/IEC 61512-1
IEC. (2013a). *IEC 61131-3, Programmable controllers - Part 3: Programming languages*. https://webstore.iec.ch/publication/4552
IEC. (2013b). *IEC 62264-1, Enterprise-control system integration, Part 1: Models and terminology 2nd ed.*
Industrial Value Chain Initiative. (2016). *Industrial Value Chain Reference Architecture (IVRA)*. https://iv-i.org/wp-test/wp-content/uploads/2017/09/doc_161208_Industrial_Value_Chain_Reference_Architecture.pdf
International Council on Systems Engineering. (2015). *INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities, 4th Edition* (D. D. Walden, G. J. Roedler, K. J. Forsberg, R. D. Hamelin, & T. M. Shortell, Eds.). Wiley. https://www.wiley.com/en-us/INCOSE+Systems+Engineering+Handbook%3A+A+Guide+for+System+Life+Cycle+Processes+and+Activities%2C+4th+Edition-p-9781118999400
International Standards Organization. (2011). *ISO/IEC 25010:2011 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models*. https://www.iso.org/standard/35733.html

International Standards Organization. (2014a). *ISO 22400-1, Automation Systems and Integration—Key Performance Indicators (KPIs) for Manufacturing Operations Management—Part 1: Overview, Concepts and Terminology*.
International Standards Organization. (2014b). *ISO 22400-2, Automation Systems and Integration—Key Performance Indicators (KPIs) for Manufacturing Operations Management—Part 2: Definitions and Descriptions*.
International Standards Organization. (2015). *ISO 9000:2015 Quality management systems — Fundamentals and vocabulary*.
ISA. (1995). *ANSI/ISA–88.01–1995, Batch Control, Part 1: Models and Terminology*. https://gmpua.com/GAMP/ISA-88.pdf
Ishikawa, K. (1990). *Introduction to Quality Control* ((Translator: J. H. Loftus), Ed.).
Jalali, A., Wohed, P., Ouyang, C., & Johannesson, P. (2013). Dynamic weaving in aspect oriented business process management. In R. Meersman (Ed.), *On the Move to Meaningful Internet Systems: Vol. 8185 LNCS* (pp. 2–20). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-41030-7_2
Janiesch, C., Koschmider, A., Mecella, M., Weber, B., Burattin, A., di Ciccio, C., Gal, A., Kannengiesser, U., Mannhardt, F., Mendling, J., Oberweis, A., Reichert, M., Rinderle-Ma, S., Song, W., Su, J., Torres, V., Weidlich, M., Weske, M., & Zhang, L. (2017). *The Internet-of-Things Meets Business Process Management: Mutual Benefits and Challenges*. 1–9. http://arxiv.org/abs/1709.03628
Jeschke, S., Brecher, C., Meisen, T., Özdemir, D., & Eschert, T. (2017). *Industrial Internet of Things and Cyber Manufacturing Systems* (pp. 3–19). Springer, Cham. https://doi.org/10.1007/978-3-319-42559-7_1
Jiang, J. R. (2018). An improved cyber-physical systems architecture for Industry 4.0 smart factories. *Advances in Mechanical Engineering*, *10*(6). https://doi.org/10.1177/1687814018784192
Johansen, K., Rao, S., & Ashourpour, M. (2021). The Role of Automation in Complexities of High-Mix in Low-Volume Production – A Literature Review. *Procedia CIRP*, *104*, 1452–1457. https://doi.org/10.1016/J.PROCIR.2021.11.245
Johansson, P. E. C., Mattsson, S., Moestam, L., & Fast-Berglund, Å. (2016). Multi-variant Truck Production - Product Variety and its Impact on Production Quality in Manual Assembly. *Procedia CIRP*, *54*, 245–250. https://doi.org/10.1016/J.PROCIR.2016.05.062
Johnsen, S. M., & Hvam, L. (2019). Understanding the impact of non-standard customisations in an engineer-to-order context: A case study. *International Journal of Production Research*, *57*(21), 6780–6794. https://doi.org/10.1080/00207543.2018.1471239
Johnsson, C. (2008). Graphical languages for business processes and manufacturing operations. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, *17*(1 PART 1). https://doi.org/10.3182/20080706-5-KR-1001.3026
Jovane, F., Koren, Y., & Boër, C. R. (2003). Present and future of flexible automation: Towards new paradigms. *CIRP Annals - Manufacturing Technology*, *52*(2), 543–560. https://doi.org/10.1016/S0007-8506(07)60203-0
Kadir, B. A., & Broberg, O. (2020). Human well-being and system performance in the transition to industry 4.0. *International Journal of Industrial Ergonomics*, *76*, 102936. https://doi.org/10.1016/J.ERGON.2020.102936
Kagermann, H., Wahlster, W., & Helbig, J. (2013). *Recommendations for implementing the strategic initiative INDUSTRIE 4.0 April 2013 Securing the future of German manufacturing industry Final report of the Industrie 4.0 Working Group*. https://www.din.de/blob/76902/e8cac883f42bf28536e7e8165993f1fd/recommendations-for-implementing-industry-4-0-data.pdf
Kamara, J. M., Anumba, C. J., & Evbuomwan, N. F. O. (2000). Process model for client requirements processing in construction. *Business Process Management Journal*, *6*(3), 251–279. https://doi.org/10.1108/14637150010325462
Kandasamy, D. (2021). *Process Oriented manufacturing execution system selection framework for small & medium-sized enterprises* [Master thesis, Eindhoven University of Technology]. https://pure.tue.nl/ws/portalfiles/portal/180267389/Master_Thesis_Dinesh_Kandasamy.pdf

Kang, N., Zhao, C., Li, J., & Horst, J. A. (2016). A Hierarchical structure of key performance indicators for operation management and continuous improvement in production systems. *International Journal of Production Research*, *54*(21), 6333–6350. https://doi.org/10.1080/00207543.2015.1136082

Kannan, S. M., Suri, K., Cadavid, J., Barosan, I., van den Brand, M., Alferez, M., & Gerard, S. (2017). Towards Industry 4.0: Gap Analysis between Current Automotive MES and Industry Standards Using Model-Based Requirement Engineering. *2017 IEEE International Conference on Software Architecture Workshops (ICSAW)*, 29–35. https://doi.org/10.1109/ICSAW.2017.53

Kannengiesser, U., Neubauer, M., & Heininger, R. (2015). Subject-oriented BPM as the glue for integrating enterprise processes in smart factories. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *9416*, 77–86. https://doi.org/10.1007/978-3-319-26138-6_11

Karaulova, T., Andronnikov, K., Mahmood, K., & Shevtshenko, & E. (2019). *Lean Automation for Low-Volume Manufacturing Environment*. 59–0068. https://doi.org/10.2507/30th.daaam.proceedings.008

Kazman, R., Bass, L., Abowd, G., & Webb, M. (1994). SAAM: a method for analyzing the properties of software architectures. *Proceedings - International Conference on Software Engineering*, 81–90. https://doi.org/10.1109/ICSE.1994.296768

Kazman, R., Klein, M., Barbacci, M., Longstaff, T., Lipson, H., & Carriere, J. (1998). The architecture tradeoff analysis method. *4th IEEE International Conference on Engineering of Complex Computer Systems, ICECCS*, 68–78. https://doi.org/10.1109/ICECCS.1998.706657

Keddis, N., Kainz, G., Brandenbourger, B., Vathoopan, M., & Zoitl, A. (2016). Handling errors in dynamic production environments. *IEEE International Conference on Industrial Informatics (INDIN)*, *0*, 148–153. https://doi.org/10.1109/INDIN.2016.7819149

Khabbazi, M. R., Hasan, M. K., Sulaiman, R., & Shapi', A. (2013). Business Process Modelling in Production Logistics: Complementary Use of BPMN and UML. *Middle-East Journal of Scientific Research*, *15*(4), 516–529. https://doi.org/10.5829/idosi.mejsr.2013.15.4.2280

Khan, A., & Turowski, K. (2016). *A Perspective on Industry 4.0: From Challenges to Opportunities in Production Systems*. https://doi.org/10.5220/0005929704410448

Khan, W. A., Raouf, A., & Cheng, K. (2011). *Augmented Reality for Manufacturing* (pp. 1–56). Springer Series in Advanced Manufacturing. https://doi.org/10.1007/978-0-85729-186-8_1

Kim, S. W., Kong, J. H., Lee, S. W., & Lee, S. (2021). Recent Advances of Artificial Intelligence in Manufacturing Industrial Sectors: A Review. *International Journal of Precision Engineering and Manufacturing*, 1–19. https://doi.org/10.1007/S12541-021-00600-3/TABLES/7

Kim, Y., & Lee, J. (1993). Manufacturing strategy and production systems: An integrated framework. *Journal of Operations Management*, *11*(1), 3–15. https://doi.org/10.1016/0272-6963(93)90029-O

Kipphan, H. (2001). *Handbook of Print Media* (H. Kipphan, Ed.). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-29900-4

Kletti, J. (2007). *Manufacturing Execution Systems-MES* (J. Kletti, Ed.). Springer-Verlag Berlin Heidelberg. https://link.springer.com/content/pdf/10.1007/978-3-540-49744-8.pdf

Knoch, S., Herbig, N., Ponpathirkoottam, S., Kosmalla, F., Staudt, P., Fettke, P., & Loos, P. (2018). Enhancing Process Data in Manual Assembly Workflows. *Lecture Notes in Business Information Processing*, *342*, 269–280. https://doi.org/10.1007/978-3-030-11641-5_21

Knoch, S., Herbig, N., Ponpathirkoottam, S., Kosmalla, F., Staudt, P., Porta, D., Fettke, P., & Loos, P. (2020). Sensor-based Human–Process Interaction in Discrete Manufacturing. *Journal on Data Semantics*, *9*(1), 21–37. https://doi.org/10.1007/S13740-019-00109-Z/FIGURES/7

Ko, R. K. L., Lee, S. S. G., & Lee, E. W. (2009). Business process management (BPM) standards: A survey. *Business Process Management Journal*, *15*(5), 744–791. https://doi.org/10.1108/14637150910987937

Koren, Y. (2006). General RMS Characteristics. Comparison with Dedicated and Flexible Systems. *Reconfigurable Manufacturing Systems and Transformable Factories*, 27–45. https://doi.org/10.1007/3-540-29397-3_3

Koren, Y. (2010). The Global Manufacturing Revolution: Product-Process-Business Integration and Reconfigurable Systems. *The Global Manufacturing Revolution: Product-Process-Business Integration and Reconfigurable Systems*. https://doi.org/10.1002/9780470618813
Koren, Y., Heisel, U., Jovane, F., Moriwaki, T., Pritschow, G., Ulsoy, G., & van Brussel, H. (1999). Reconfigurable manufacturing systems. *CIRP Annals - Manufacturing Technology*, *48*(2), 527–540. https://doi.org/10.1016/S0007-8506(07)63232-6
Koren, Y., & Shpitalni, M. (2010). Design of reconfigurable manufacturing systems. *Journal of Manufacturing Systems*, *29*(4), 130–141. https://doi.org/10.1016/j.jmsy.2011.01.001
Kowalkowski, C. (2011). Dynamics of value propositions: insights from service-dominant logic. *European Journal of Marketing*, *45*(1/2), 277–294. https://doi.org/10.1108/03090561111095702
Kruchten, P. B. (1995). The 4+1 View Model of Architecture. *IEEE Software*, *12*(6), 42–50. https://doi.org/10.1109/52.469759
Krumeich, J., Weis, B., Werth, D., & Loos, P. (2014). Event-Driven Business Process Management: where are we now? A comprehensive synthesis and analysis of literature. *Business Process Management Journal*, *20*(4), 615–633. https://doi.org/10.1108/BPMJ-07-2013-0092
Kuo, T. C., Hsu, C. W., Ku, K. C., Chen, P. S., & Lin, C. H. (2012). A collaborative model for controlling the green supply network in the motorcycle industry. *Advanced Engineering Informatics*, *26*(4), 941–950. https://doi.org/10.1016/J.AEI.2012.09.001
Kurbel, K. E. (2013). *Enterprise Resource Planning and Supply Chain Management*. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-31573-2
Kurz, M., Fleischmann, A., Lederer, M., & Huber, S. (2013). Planning for the Unexpected: Exception Handling and BPM. *Communications in Computer and Information Science*, *360 CCIS*, 123–149. https://doi.org/10.1007/978-3-642-36754-0_8
Kushalnagar, N., Montenegro, G., & Schumacher, C. P. P. (2007). IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals. *Undefined*. https://doi.org/10.17487/RFC4919
la Rosa, M., ter Hofstede, A. H. M., Wohed, P., Reijers, H. A., Mendling, J., & van der Aalst, W. M. P. (2011). Managing process model complexity via concrete syntax modifications. *IEEE Transactions on Industrial Informatics*, *7*(2), 255–265. https://doi.org/10.1109/TII.2011.2124467
Lampel, J., & Mintzberg, H. (1996). Customizing Customization. *Sloan Management Review*, *38*(1), 21–30. https://www.researchgate.net/publication/40962226
Landers, R. G., Min, B. K., & Koren, Y. (2001). Reconfigurable machine tools. *CIRP Annals - Manufacturing Technology*, *50*(1), 269–274. https://doi.org/10.1016/S0007-8506(07)62120-9
Lanza, J., Sánchez, L., Gómez, D., Santana, J. R., & Sotres, P. (2019). A Semantic-Enabled Platform for Realizing an Interoperable Web of Things. *Sensors 2019, Vol. 19, Page 869*, *19*(4), 869. https://doi.org/10.3390/S19040869
Lasi, H., Fettke, P., Kemper, H.-G., Feld, T., & Hoffmann, M. (2014). Industry 4.0. *Business & Information Systems Engineering*, *6*(4), 239–242. https://doi.org/10.1007/s12599-014-0334-4
Le-Anh, T., & de Koster, M. B. M. (2006). A review of design and control of automated guided vehicle systems. *European Journal of Operational Research*, *171*(1), 1–23. https://doi.org/10.1016/j.ejor.2005.01.036
Lee, J. (2015). Smart Factory Systems. *Informatik-Spektrum*, *38*(3), 230–235. https://doi.org/10.1007/s00287-015-0891-z
Lee, J., Bagheri, B., & Kao, H. A. (2015). A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems. *Manufacturing Letters*, *3*, 18–23. https://doi.org/10.1016/j.mfglet.2014.12.001
Lee, J., Kao, H. A., & Yang, S. (2014). Service Innovation and Smart Analytics for Industry 4.0 and Big Data Environment. *Procedia CIRP*, *16*, 3–8. https://doi.org/10.1016/J.PROCIR.2014.02.001
Leitão, P. (2009). Agent-based distributed manufacturing control: A state-of-the-art survey. *Engineering Applications of Artificial Intelligence*, *22*(7), 979–991. https://doi.org/10.1016/J.ENGAPPAI.2008.09.005

Leitner, M., Mangler, J., & Rinderle-Ma, S. (2012). Definition and Enactment of Instance-Spanning Process Constraints. In X. S. Wang, I. Cruz, A. Delis, & G. Huang (Eds.), *Web Information Systems Engineering - WISE 2012: Vol. 7651 LNCS* (pp. 652–658). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-35063-4_49
Leitner, M., & Rinderle-Ma, S. (2014). A systematic review on security in Process-Aware Information Systems – Constitution, challenges, and future directions. *Information and Software Technology*, *56*(3), 273–293. https://doi.org/10.1016/J.INFSOF.2013.12.004
Lerner, B. S., Christov, S., Osterweil, L. J., Bendraou, R., Kannengiesser, U., & Wise, A. (2010). Exception handling patterns for process modeling. *IEEE Transactions on Software Engineering*, *36*(2), 162–183. https://doi.org/10.1109/TSE.2010.1
Li, Q., Tang, Q., Chan, I., Wei, H., Pu, Y., Jiang, H., Li, J., & Zhou, J. (2018). Smart manufacturing standardization: Architectures, reference models and standards framework. *Computers in Industry*, *101*, 91–106. https://doi.org/10.1016/j.compind.2018.06.005
Lichtblau, K., Stich, V., Bertenrath, R., Blum, M., Bleider, M., Millack, A., Schmitt, K., Schmitz, E., & Schröter, M. (2015). *IMPULS INDUSTRIE 4.0 READINESS*. http://industrie40.vdma.org/documents/4214230/5356229/Industrie 4.0 Readiness Study English.pdf/f6de92c1-74ed-4790-b6a4-74b30b1e83f0
Lin, S.-W., Miller, B., Durand, J., Bleakley, G., Chigani, A., Martin, R., Murphy, B., & Crawford, M. (2017). *The Industrial Internet of Things Volume G1: Reference Architecture*. https://www.iiconsortium.org/IIC_PUB_G1_V1.80_2017-01-31.pdf
Lindorfer, R., Froschauer, R., & Schwarz, G. (2018). ADAPT - A decision-model-based Approach for Modeling Collaborative Assembly and Manufacturing Tasks. *Proceedings - IEEE 16th International Conference on Industrial Informatics, INDIN 2018*, 559–564. https://doi.org/10.1109/INDIN.2018.8472064
Lindsay, A., Downs, D., & Lunn, K. (2003). Business processes - Attempts to find a definition. *Information and Software Technology*, *45*(15), 1015–1019. https://doi.org/10.1016/S0950-5849(03)00129-0
Liu, J., & Liu, P. (2010). Status and key techniques in cloud computing. *ICACTE 2010 - 2010 3rd International Conference on Advanced Computer Theory and Engineering, Proceedings*, *4*(1), 285–288. https://doi.org/10.1109/ICACTE.2010.5579728
Lohmann, N., & Nyolt, M. (2011). Artifact-Centric Modeling Using BPMN. *Service-Oriented Computing - ICSOC 2011 Workshops. ICSOC 2011*, *7221 LNCS*, 54–65. https://doi.org/10.1007/978-3-642-31875-7_7
Lohmann, N., & Wolf, K. (2010). Artifact-Centric Choreographies. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *6470 LNCS*, 32–46. https://doi.org/10.1007/978-3-642-17358-5_3
Lohmeyer, B. (2013). *Writing Use Cases: Exception or Alternate Flow?* https://www.lohmy.de/post/writing-use-cases-exception-or-alternate-flow
Longo, F., Nicoletti, L., & Padovano, A. (2017). Smart operators in industry 4.0: A human-centered approach to enhance operators' capabilities and competencies within the new smart factory context. *Computers and Industrial Engineering*, *113*, 144–159. https://doi.org/10.1016/j.cie.2017.09.016
Lu, Y. (2017). Industry 4.0: A survey on technologies, applications and open research issues. *Journal of Industrial Information Integration*, *6*, 1–10. https://doi.org/10.1016/J.JII.2017.04.005
Lu, Y., & Cecil, J. (2015). An Internet of Things (IoT)-based collaborative framework for advanced manufacturing. *The International Journal of Advanced Manufacturing Technology*. https://doi.org/10.1007/s00170-015-7772-0
Lu, Y., Morris, K., & Frechette, S. (2016). *Current Standards Landscape for Smart Manufacturing Systems*. https://doi.org/10.6028/NIST.IR.8107
Luck, M. (1995). *A Formal Framework for Agency and Autonomy Mark d'Inverno*. www.aaai.org
Ludewig, J. (2003). Models in software engineering ? an introduction. *Software and Systems Modeling*, *2*(1), 5–14. https://doi.org/10.1007/s10270-003-0020-3
Lunenburg, F. C. (2012). Organizational Structure Mintzberg's Framework. *International Journal of Scholarly, Academic, Intellectual Diversity*, *14*(1).

https://platform.europeanmoocs.eu/users/8/Lunenburg-Fred-C.-Organizational-Structure-Mintzberg-Framework-IJSAID-V14-N1-2012.pdf
Luo, Z., Sheth, A., Kochut, K., & Miller, J. (2000). Exception Handling in Workflow Systems. *Applied Intelligence 2000 13:2*, *13*(2), 125–147. https://doi.org/10.1023/A:1008388412284
Luthra, S., & Mangla, S. K. (2018). Evaluating challenges to Industry 4.0 initiatives for supply chain sustainability in emerging economies. *Process Safety and Environmental Protection*, *117*, 168–179. https://doi.org/10.1016/j.psep.2018.04.018
Mahdavi, M., & Olsen, T. L. (2017). Designing Effective Supply Chains in Strategic Alignment with Demand Characteristics and Market Requirements. *Foundations and Trends® in Technology, Information and Operations Management*, *10*(2), 89–204. https://doi.org/10.1561/0200000049
Mankins, J. C. (2009). Technology readiness assessments: A retrospective. *Acta Astronautica*, *65*(9–10), 1216–1223. https://doi.org/10.1016/j.actaastro.2009.03.058
Mantravadi, S., & Møller, C. (2019). An Overview of Next-generation Manufacturing Execution Systems: How important is MES for Industry 4.0? *Procedia Manufacturing*, *30*, 588–595. https://doi.org/10.1016/J.PROMFG.2019.02.083
March, S. T., & Smith, G. F. (1995). Design and natural science research on information technology. *Decision Support Systems*, *15*(4), 251–266. https://doi.org/10.1016/0167-9236(94)00041-2
Marengo, E., Nutt, W., & Perktold, M. (2018). Construction process modeling: Representing activities, items and their interplay. In M. Weske, M. Montali, I. Weber, & J. vom Brocke (Eds.), *Business Process Management. BPM 2018: Vol. 11080 LNCS* (pp. 48–65). Springer Verlag. https://doi.org/10.1007/978-3-319-98648-7_4/TABLES/1
Mařík, V., & McFarlane, D. (2005). Industrial adoption of agent-based technologies. *IEEE Intelligent Systems*, *20*(1), 27–35. https://doi.org/10.1109/MIS.2005.11
Marrella, A. (2019). Automated Planning for Business Process Management. *Journal on Data Semantics*, *8*(2), 79–98. https://doi.org/10.1007/S13740-018-0096-0/FIGURES/6
Marrella, A., Mecella, M., & Sardiña, S. (2018). Supporting adaptiveness of cyber-physical processes through action-based formalisms. *AI Communications*, *31*(1), 47–74. https://doi.org/10.3233/AIC-170748
Mavin, A., & Wilkinson, P. (2010). BIG EARS (The return of "Easy Approach to Requirements Syntax"). *Proceedings of the 2010 18th IEEE International Requirements Engineering Conference, RE2010*, 277–282. https://doi.org/10.1109/RE.2010.39
Mavin, A., Wilkinson, P., Harwood, A., & Novak, M. (2009). EARS (Easy Approach to Requirements Syntax). *Proceedings of the IEEE International Conference on Requirements Engineering*, 317–322. https://doi.org/10.1109/RE.2009.9
Mazmela, M., Lasa, G., Aranburu, E., Gonzalez, I., & Reguera, D. (2018). TAMUX model for industrial HMI evaluation from UX and task performance perspective. *Proceedings of the XIX International Conference on Human Computer Interaction*. https://doi.org/10.1145/3233824
McKinsey Digital. (2015). *Industry 4.0 How to navigate digitization of the manufacturing sector*. https://www.mckinsey.com/~/media/McKinsey/Business Functions/Operations/Our Insights/Industry 40 How to navigate digitization of the manufacturing sector/Industry-40-How-to-navigate-digitization-of-the-manufacturing-sector.ashx
Mehandjiev, Nikolay., & Grefen, Paul. (2010). *Dynamic business process formation for instant virtual enterprises*. Springer.
Mehrabi, M. G., Ulsoy, A. G., & Koren, Y. (2000). Reconfigurable manufacturing systems: Key to future manufacturing. *Journal of Intelligent Manufacturing 2000 11:4*, *11*(4), 403–419. https://doi.org/10.1023/A:1008930403506
Mehta, B. R., & Reddy, Y. J. (2015). Batch automation systems. *Industrial Process Automation Systems*, 135–156. https://doi.org/10.1016/B978-0-12-800939-0.00003-6
Mendling, J., van Dongen, B. F., & van der Aalst, W. M. P. (2008). Getting rid of OR-joins and multiple start events in business process models. *Enterprise Information Systems*, *2*(4), 403–419. https://doi.org/10.1080/17517570802245433
MESA. (1997). *MES Functionalities & MRP to MES Data Flow Possibilities*.
MESA. (2014). *Manufacturing Metrics that Really Matter*. https://mesa.org/topics-resources/manufacturing-analytics/analytics-that-matter/

Meyer, A., Herzberg, N., Puhlmann, F., & Weske, M. (2014). Implementation Framework for Production Case Management: Modeling and Execution. *Proceedings . IEEE 18th International Enterprise Distributed Object Computing Conference*, *2014-December*(December), 190–199. https://doi.org/10.1109/EDOC.2014.34
Meyer, A., Kruse, K., Lindhauer, S., Weske, T., Meyer, A., Pufahl, L., Batoulis, K., Kruse, S., Lindhauer, T., Stoff, T., Fahland, D., & Weske, M. (2013). Data perspective in process choreographies : modeling and execution. *BPM Reports*. www.tue.nl/taverne
Meyer, A., Pufahl, L., Batoulis, K., Fahland, D., & Weske, M. (2015). Automating data exchange in process choreographies. *Information Systems*, *53*, 296–329. https://doi.org/10.1016/j.is.2015.03.008
Meyer, A., & Weske, M. (2014). Activity-centric and artifact-centric process model roundtrip. *Lecture Notes in Business Information Processing*, *171 171 LNBIP*, 167–181. https://doi.org/10.1007/978-3-319-06257-0_14
Meyer, S., Ruppen, A., & Hilty, L. (2015). The things of the internet of things in BPMN. *Lecture Notes in Business Information Processing*, *215*, 285–297. https://doi.org/10.1007/978-3-319-19243-7_27
Meyer, S., Ruppen, A., & Magerkurth, C. (2013). Internet of Things-Aware Process Modeling: Integrating IoT Devices as Business Process Resources. In C. Salinesi, M. C. Norrie, & Ó. Pastor (Eds.), *Advanced Information Systems Engineering. CAiSE 2013: Vol. 7908 LNCS* (pp. 84–98). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-38709-8_6
Michalik, P., Štofa, J., & Zolotová, I. (2013). The Use of Bpmn for Modelling The Mes Level in Information and Control Systems. *Quality Innovation Prosperity*, *17*(1), 39–47. https://doi.org/10.12776/qip.v17i1.68
Michalos, G., Makris, S., Papakostas, N., Mourtzis, D., & Chryssolouris, G. (2010). Automotive assembly technologies review: challenges and outlook for a flexible and adaptive approach. In *CIRP Journal of Manufacturing Science and Technology* (Vol. 2, Issue 2, pp. 81–91). https://doi.org/10.1016/j.cirpj.2009.12.001
MIIT, & SAC. (2015). *National Intelligent Manufacturing Standard System Construction Guidelines*.
Miller, D., & Friesen, P. H. (1983). Strategy-Making and Environment: The Third Link. In *Management Journal* (Vol. 4, Issue 3). https://www.jstor.org/stable/2486083
Miller, D., & Shamsie, J. (1999). Strategic responses to three kinds of uncertainty: Product line simplicity at the Hollywood film studios. *Journal of Management*, *25*(1), 97–116. https://doi.org/10.1016/S0149-2063(99)80005-1
Milliken, A. (2011). Using Exception Management to Improve the Demand Forecast. *The Journal of Business Forecasting*, *30*(1), 4.
Miqueo, A., Torralba, M., & Yagüe-Fabra, J. A. (2020). Lean Manual Assembly 4.0: A Systematic Review. *Applied Sciences 2020, Vol. 10, Page 8555*, *10*(23), 8555. https://doi.org/10.3390/APP10238555
Mishra, R., Pundir, A. K., & Ganapathy, L. (2014). Manufacturing flexibility research: A review of literature and agenda for future research. In *Global Journal of Flexible Systems Management* (Vol. 15, Issue 2, pp. 101–112). Global Institute of Flexible Systems Management. https://doi.org/10.1007/s40171-013-0057-2
Mittal, S., Khan, M. A., Romero, D., & Wuest, T. (2018). A critical review of smart manufacturing & Industry 4.0 maturity models: Implications for small and medium-sized enterprises (SMEs). *Journal of Manufacturing Systems*, *49*, 194–214. https://doi.org/10.1016/J.JMSY.2018.10.005
Moeuf, A., Pellerin, R., Lamouri, S., Tamayo-Giraldo, S., & Barbaray, R. (2017). The industrial management of SMEs in the era of Industry 4.0. *Https://Doi.Org/10.1080/00207543.2017.1372647*, *56*(3), 1118–1136. https://doi.org/10.1080/00207543.2017.1372647
Molnar, M. (2015). *The National Network for Manufacturing Innovation. A U.S. Initiative for Sustainable Collaboration NSF Workshop on Sustainable Manufacturing: Urgent Research Needs and Multidisciplinary Collaboration*. http://www.research.che.utexas.edu/susman/documents/aiche2015/2015 NSF SM Workshop_Keynote_DOC-NNMI M Molner_082015.pdf

Monk, E. F., & Wagner, B. J. (2013). *Concepts in enterprise resource planning*. 254.
Monostori, L. (2014). Cyber-physical production systems: Roots, expectations and R&D challenges. *Procedia CIRP*, *17*, 9–13. https://doi.org/10.1016/j.procir.2014.03.115
Monostori, L., Váncza, J., & Kumara, S. R. T. (2006). Agent-Based Systems for Manufacturing. *CIRP Annals*, *55*(2), 697–720. https://doi.org/10.1016/J.CIRP.2006.10.004
Moody, D. L. (2003). The Method Evaluation Model: A TheoreticalModel for Validating Information Systems DesignMethods. *ECIS 2003 Proceedings*. http://aisel.aisnet.org/ecis2003/79
Moreira, A., Rashid, A., & Araújc, J. (2005). Multi-dimensional separation of concerns in requirements engineering. *Proceedings of the IEEE International Conference on Requirements Engineering*, 285–296. https://doi.org/10.1109/RE.2005.46
Mosterman, P. J., & Zander, J. (2016). Cyber-physical systems challenges: a needs analysis for collaborating embedded software systems. *Software & Systems Modeling*, *15*(1), 5–16. https://doi.org/10.1007/s10270-015-0469-x
Motahari-Nezhad, H. R., & Swenson, K. D. (2013). Adaptive case management: Overview and research challenges. *Proceedings - 2013 IEEE International Conference on Business Informatics, IEEE CBI 2013*, 264–269. https://doi.org/10.1109/CBI.2013.44
Mourão, H., & Antunes, P. (2005). A collaborative framework for unexpected exception handling. In H. Fukś, S. Lukosch, & A. C. Salgado (Eds.), *Groupware: Design, Implementation, and Use. CRIWG 2005: Vol. 3706 LNCS* (pp. 168–183). Springer, Berlin, Heidelberg. https://doi.org/10.1007/11560296_13
Mourtzis, D., Fotia, S., Boli, N., & Pittaro, P. (2018). Product-service system (PSS) complexity metrics within mass customization and Industry 4.0 environment. *The International Journal of Advanced Manufacturing Technology*, *97*(1–4), 91–103. https://doi.org/10.1007/s00170-018-1903-3
Mourtzis, D., Fotia, S., Boli, N., & Vlachou, E. (2019). Modelling and quantification of industry 4.0 manufacturing complexity based on information theory: a robotics case study. *International Journal of Production Research*, *57*(22), 6908–6921. https://doi.org/10.1080/00207543.2019.1571686
Mpofu, T. P., Mawere, C., & Mukosera, M. (2014). The Impact and Application of 3D Printing Technology Article in. In *International Journal of Science and Research*. https://www.researchgate.net/publication/291975129
Muhammad, U., Ferrer, B. R., Mohammed, W. M., & Lastra, J. L. M. (2018). An approach for implementing key performance indicators of a discrete manufacturing simulator based on the ISO 22400 standard. *Proceedings - 2018 IEEE Industrial Cyber-Physical Systems, ICPS 2018*, 629–636. https://doi.org/10.1109/ICPHYS.2018.8390779
Mussbacher, G., Amyot, D., Breu, R., Bruel, J.-M., Cheng, B. H. C., Collet, P., Combemale, B., France, R. B., Heldal, R., Hill, J., Kienzle, J., Schöttle, M., Steimann, F., Stikkolorum, D., & Whittle, J. (2014). The Relevance of Model-Driven Engineering Thirty Years from Now. In *LNCS* (Vol. 8767, pp. 183–200). https://doi.org/10.1007/978-3-319-11653-2_12
Nagorny, K., Colombo, A. W., & Schmidtmann, U. (2012). A service- and multi-agent-oriented manufacturing automation architecture: An IEC 62264 level 2 compliant implementation. *Computers in Industry*, *63*(8), 813–823. https://doi.org/10.1016/J.COMPIND.2012.08.003
Nahmias, S., & Lennon Olsen, T. (2015). Production and operations analysis. In *Waveland Press, Inc.* (Seventh edition.). Waveland Press Inc.
Napoleone, A., Macchi, M., & Pozzetti, A. (2020). A review on the characteristics of cyber-physical systems for the future smart factories. *Journal of Manufacturing Systems*, *54*, 305–335. https://doi.org/10.1016/J.JMSY.2020.01.007
Nee, A. Y. C., Ong, S. K., Chryssolouris, G., & Mourtzis, D. (2012). Augmented reality applications in design and manufacturing. *CIRP Annals*, *61*(2), 657–679. https://doi.org/10.1016/J.CIRP.2012.05.010
Neubauer, M., & Stary, C. (2016). S-BPM in the production industry: A stakeholder approach. In *S-BPM in the Production Industry: A Stakeholder Approach*. https://doi.org/10.1007/978-3-319-48466-2

Newman, S. T., Nassehi, A., Xu, X. W., Rosso, R. S. U., Wang, L., Yusof, Y., Ali, L., Liu, R., Zheng, L. Y., Kumar, S., Vichare, P., & Dhokia, V. (2008). Strategic advantages of interoperability for global manufacturing using CNC technology. *Robotics and Computer-Integrated Manufacturing*, *24*(6), 699–708. https://doi.org/10.1016/j.rcim.2008.03.002

Ngo, T. D., Kashani, A., Imbalzano, G., Nguyen, K. T. Q., & Hui, D. (2018). Additive manufacturing (3D printing): A review of materials, methods, applications and challenges. In *Composites Part B: Engineering* (Vol. 143, pp. 172–196). Elsevier Ltd. https://doi.org/10.1016/j.compositesb.2018.02.012

Nitsche, B., & Straube, F. (2020). Efficiently managing supply chain volatility - A management framework for the manufacturing industry. *Procedia Manufacturing*, *43*, 320–327. https://doi.org/10.1016/j.promfg.2020.02.166

Nüttgens, M., Feld, T., & Zimmermann, V. (1998). Business Process Modeling with EPC and UML: Transformation or Integration? *The Unified Modeling Language*, 250–261. https://doi.org/10.1007/978-3-642-48673-9_17

Nwankpa, J. K. (2015). ERP system usage and benefit: A model of antecedents and outcomes. *Computers in Human Behavior*, *45*, 335–344. https://doi.org/10.1016/J.CHB.2014.12.019

O'Kane, J. M. (2014). *A gentle introduction to ROS*. Jason M. O'Kane. https://www.cse.sc.edu/~jokane/agitr/agitr-letter.pdf

O'Leary-Kelly, S. W., & Flores, B. E. (2002). The integration of manufacturing and marketing/sales decisions: impact on organizational performance. *Journal of Operations Management*, *20*(3), 221–240. https://doi.org/10.1016/S0272-6963(02)00005-0

Olechowski, A., Eppinger, S. D., & Joglekar, N. (2015). Technology readiness levels at 40: A study of state-of-the-art use, challenges, and opportunities. *Portland International Conference on Management of Engineering and Technology*, *2015-Septe*, 2084–2094. https://doi.org/10.1109/PICMET.2015.7273196

Olhager, J. (2003). Strategic positioning of the order penetration point. *International Journal of Production Economics*, *85*(3), 319–329. https://doi.org/10.1016/S0925-5273(03)00119-1

ORACLE. (2017). *JD Edwards World Product Data Management - Discrete Guide: Work with Work Centers*. https://docs.oracle.com/cd/E26228_01/doc.93/e21777/ch_wk_wk_cntrs.htm#WEAPD337

Orzes, G., Poklemba, R., & Towner, W. T. (2020). Implementing Industry 4.0 in SMEs: A Focus Group Study on Organizational Requirements. *Industry 4.0 for SMEs: Challenges, Opportunities and Requirements*, 251–277. https://doi.org/10.1007/978-3-030-25425-4_9

Osterrieder, P., Budde, L., & Friedli, T. (2020). The smart factory as a key construct of industry 4.0: A systematic literature review. *International Journal of Production Economics*, *221*. https://doi.org/10.1016/j.ijpe.2019.08.011

Oztemel, E., & Gursev, S. (2020). Literature review of Industry 4.0 and related technologies. *Journal of Intelligent Manufacturing*, *31*(1), 127–182. https://doi.org/10.1007/s10845-018-1433-8

Palmarini, R., Erkoyuncu, J. A., & Roy, R. (2017). An Innovative Process to Select Augmented Reality (AR) Technology for Maintenance. *Procedia CIRP*, *59*, 23–28. https://doi.org/10.1016/J.PROCIR.2016.10.001

Palmer, N. (2009). Workflow Management Coalition. *Encyclopedia of Database Systems*, 3550–3550. https://doi.org/10.1007/978-0-387-39940-9_472

Pantano, M., Blumberg, A., Regulin, D., Hauser, T., & Lee, D. (2021). *Design of a collaborative modular end effector consider-ing human values and safety requirements for industrial use cases*.

Pantano, M., Pavlovskyi, Y., Schulenburg, E., Traganos, K., Ahmadi, S., Regulin, D., Lee, D., Saenz, J., Pini, F., Francalanza, E., & Fraboni, F. (2022). Novel Approach using Risk Analysis Component to Continuously Update Collaborative Robotics Applications in the Smart, Connected Factory Model. *Applied Sciences 2022, Vol. 12, Page 5639*, *12*(11), 5639. https://doi.org/10.3390/APP12115639

Park, H. S., & Tran, N. H. (2012). An autonomous manufacturing system based on swarm of cognitive agents. *Journal of Manufacturing Systems*, *31*(3), 337–348. https://doi.org/10.1016/J.JMSY.2012.05.002

Park, K., & Okudan Kremer, G. E. (2015). Assessment of static complexity in design and manufacturing of a product family and its impact on manufacturing performance. *International Journal of Production Economics*, *169*, 215–232. https://doi.org/10.1016/j.ijpe.2015.07.036

Parmenter, D. (2010). *Key Performance Indicator Developing, Implementing, and Using Winning KPIs. Second Edition*. John Wiley & Sons, Inc.

Pauker, F., Mangler, J., Rinderle-Ma, S., & Pollak, C. (2018). Centurio.work - Modular secure manufacturing orchestration. *CEUR Workshop Proceedings*, *2196*, 164–171.

Pauls, K., McCulloch, S., Hall, R. S., & Savage, D. (2011). OSGi in Action. *Manning*, 548. http://books.google.com/books?id=y3lPPgAACAAJ&pgis=1

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, *24*(3), 45–77. https://doi.org/10.2753/MIS0742-1222240302

Pereira, A. C., & Romero, F. (2017). A review of the meanings and the implications of the Industry 4.0 concept. *Procedia Manufacturing*, *13*, 1206–1214. https://doi.org/10.1016/j.promfg.2017.09.032

Pereira, R. M., Szejka, A. L., & Canciglieri, O. (2020). A Discussion on Current Issues for Semantic Interoperability in an Integrated Manufacturing System. *Advances in Transdisciplinary Engineering*, *12*, 514–523. https://doi.org/10.3233/ATDE200112

Pérez, F., Irisarri, E., Orive, D., Marcos, M., & Estevez, E. (2015). A CPPS Architecture approach for Industry 4.0. *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*, *2015-Octob*, 4–7. https://doi.org/10.1109/ETFA.2015.7301606

Pesic, M., Schonenberg, H., & van der Aalst, W. M. P. (2007). DECLARE: Full support for loosely-structured processes. *Proceedings - IEEE International Enterprise Distributed Object Computing Workshop, EDOC*, 287–298. https://doi.org/10.1109/EDOC.2007.4384001

Pesic, M., & van der Aalst, W. M. P. (2006). A declarative approach for flexible business processes management. In J. Eder & S. Dustdar (Eds.), *Business Process Management Workshops: Vol. 4103 LNCS* (pp. 169–180). Springer, Berlin, Heidelberg. https://doi.org/10.1007/11837862_18

Petrasch, R., & Hentschke, R. (2016). Process modeling for industry 4.0 applications: Towards an industry 4.0 process modeling language and method. *2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, *Cc*, 1–5. https://doi.org/10.1109/JCSSE.2016.7748885

Pichler, P., Weber, B., Zugal, S., Pinggera, J., Mendling, J., & Reijers, H. A. (2011). Imperative versus Declarative Process Modeling Languages: An Empirical Investigation. *Lecture Notes in Business Information Processing*, *99 LNBIP*(PART 1), 383–394. https://doi.org/10.1007/978-3-642-28108-2_37

Pinarbaşi, M., Sel, Ç., Alağaş, H. M., & Yüzükirmizi, M. (2013). Integrated definition modeling and Taguchi analysis of flexible manufacturing systems: aircraft industry application. *The International Journal of Advanced Manufacturing Technology 2013 68:9*, *68*(9), 2169–2183. https://doi.org/10.1007/S00170-013-4829-9

Polderdijk, M., Vanderfeesten, I., Erasmus, J., Traganos, K., Bosch, T., Rhijn, G. van, & Fahland, D. (2017). A Visualization of Human Physical Risks in Manufacturing Processes Using BPMN. In E. Teniente & M. Weidlich (Eds.), *Business Process Management Workshops. BPM 2017.* (Vol. 308, pp. 732–743). Springer, Cham. https://doi.org/10.1007/978-3-319-74030-0_58

Pourmirza, S., Peters, S., Dijkman, R., & Grefen, P. (2017). A systematic literature review on the architecture of business process management systems. In *Information Systems* (Vol. 66, pp. 43–58). Elsevier Ltd. https://doi.org/10.1016/j.is.2017.01.007

Pourmirza, S., Peters, S., Dijkman, R., & Grefen, P. (2019). BPMS-RA: A Novel Reference Architecture for Business Process Management Systems. *ACM Transactions on Internet Technology*, *19*(1), 1–23. https://doi.org/10.1145/3232677

Prades, L., Romero, F., Estruch, A., García-Dominguez, A., & Serrano, J. (2013). Defining a methodology to design and implement business process models in BPMN according to the standard ANSI/ISA-95 in a manufacturing enterprise. *Procedia Engineering*, *63*, 115–122. https://doi.org/10.1016/j.proeng.2013.08.283

Prat, N., Comyn-Wattiau, I., & Akoka, J. (2015). A Taxonomy of Evaluation Methods for Information Systems Artifacts. *Journal of Management Information Systems*, *32*(3), 229–267. https://doi.org/10.1080/07421222.2015.1099390/SUPPL_FILE/MMIS_A_1099390_SM6514.PDF

Pufahl, L., & Weske, M. (2019). Batch activity: enhancing business process modeling and enactment with batch processing. *Computing*, *101*(12), 1909–1933. https://doi.org/10.1007/S00607-019-00717-4/FIGURES/11

Qi, Q., & Tao, F. (2019). A Smart Manufacturing Service System Based on Edge Computing, Fog Computing, and Cloud Computing. *IEEE Access*, *7*, 86769–86777. https://doi.org/10.1109/ACCESS.2019.2923610

Qian, C., Sun, W., & Zhang, Y. (2020). ResourceNet: a collaboration network among decentralised manufacturing resources for autonomous exception-handling in smart manufacturing. *IET Collaborative Intelligent Manufacturing*, *2*(3), 109–114. https://doi.org/10.1049/iet-cim.2019.0066

Qin, J., Liu, Y., & Grosvenor, R. (2016). A Categorical Framework of Manufacturing for Industry 4.0 and beyond. *Procedia CIRP*, *52*, 173–178. https://doi.org/10.1016/j.procir.2016.08.005

Radziwon, A., Bilberg, A., Bogers, M., & Madsen, E. S. (2014). The Smart Factory: Exploring Adaptive and Flexible Manufacturing Solutions. *Procedia Engineering*, *69*, 1184–1190. https://doi.org/10.1016/j.proeng.2014.03.108

Rai, R., Tiwari, M. K., Ivanov, D., & Dolgui, A. (2021). Machine learning in manufacturing and industry 4.0 applications. *Https://Doi.Org/10.1080/00207543.2021.1956675*, *59*(16), 4773–4778. https://doi.org/10.1080/00207543.2021.1956675

Raj, A., Dwivedi, G., Sharma, A., Lopes de Sousa Jabbour, A. B., & Rajak, S. (2020). Barriers to the adoption of industry 4.0 technologies in the manufacturing sector: An inter-country comparative perspective. *International Journal of Production Economics*, *224*. https://doi.org/10.1016/J.IJPE.2019.107546

Rashid, M. A., Hossain, L., & Patrick, J. D. (2002). The Evolution of ERP Systems. In *Enterprise Resource Planning: Solutions and Management* (pp. 35–50). IGI Global. https://doi.org/10.4018/978-1-930708-36-5.ch003

Reichert, M. (2011). What BPM technology can do for healthcare process support. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *6747 LNAI*, 2–13. https://doi.org/10.1007/978-3-642-22218-4_2

Reichert, M., & Weber, B. (2012a). Ad hoc Changes of Process Instances. In *Enabling Flexibility in Process-Aware Information Systems* (pp. 153–217). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-30409-5_7

Reichert, M., & Weber, B. (2012b). Exception Handling. In *Enabling Flexibility in Process-Aware Information Systems* (pp. 127–151). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-30409-5_6

Reid, I., Bamford, D., & Ismail, H. (2018). Reconciling engineer-to-order uncertainty by supporting front-end decision-making. *Https://Doi.Org/10.1080/00207543.2018.1552370*, *57*(21), 6856–6874. https://doi.org/10.1080/00207543.2018.1552370

Ren, S., Zhang, Y., Liu, Y., Sakao, T., Huisingh, D., & Almeida, C. M. V. B. (2019). A comprehensive review of big data analytics throughout product lifecycle to support sustainable smart manufacturing: A framework, challenges and future research directions. *Journal of Cleaner Production*, *210*, 1343–1365. https://doi.org/10.1016/J.JCLEPRO.2018.11.025

Reniers, G. (2017). On the future of safety in the manufacturing industry. *Procedia Manufacturing*, *13*, 1292–1296. https://doi.org/10.1016/j.promfg.2017.09.057

Rerup Schlichter, B., & Kraemmergaard, P. (2010). A comprehensive literature review of the ERP research field over a decade. *Journal of Enterprise Information Management*, *23*(4), 486–520. https://doi.org/10.1108/17410391011061780

Rescher, N. (1996). *Process Metaphysics: An Introduction to Process Philosophy*. https://philpapers.org/rec/RESPMA

Reyes, R. M., de la Riva, J., Maldonado, A., Woocay, A., & de la O, R. (2015). Association between Human Error and Occupational Accidents' Contributing Factors for Hand Injuries in the

Automotive Manufacturing Industry. *Procedia Manufacturing*, *3*, 6498–6504. https://doi.org/10.1016/J.PROMFG.2015.07.936

Rinaldi, J. (2016). *OPC UA -Unified Architecture: The Everyman's Guide to the Most Important Information Technology in Industrial Automation*. https://www.academia.edu/41900561/OPC_UA_Unified_Architecture_The_Everymans_Guide_to_the_Most_Important_Information_Technology_in_Industrial_Automation_by_John_S_Rinaldi

Rinderle, S., & Reichert, M. (2006). Data–Driven Process Control and Exception Handling in Process Management Systems. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 4001 LNCS* (pp. 273–287). Springer Verlag. https://doi.org/10.1007/11767138_19

Ritter, D., & Sosulski, J. (2014). Modeling Exception Flows in Integration Systems. *Proceedings . IEEE 18th International Enterprise Distributed Object Computing Conference*, *2014-Decem*(December), 12–21. https://doi.org/10.1109/EDOC.2014.13

Ritter, D., & Sosulski, J. (2016). Exception Handling in Message-Based Integration Systems and Modeling Using BPMN. *International Journal of Cooperative Information Systems*, *25*(2), 1650004. https://doi.org/10.1142/S0218843016500040

Rodrigues Da Silva, A. (2015). Model-driven engineering: A survey supported by the unified conceptual model. *Computer Languages, Systems & Structures*, *43*, 139–155. https://doi.org/10.1016/J.CL.2015.06.001

Rolón, M., & Martínez, E. (2012). Agent-based modeling and simulation of an autonomic manufacturing execution system. *Computers in Industry*, *63*(1), 53–78. https://doi.org/10.1016/J.COMPIND.2011.10.005

Romero, D., Bernus, P., Noran, O., Stahre, J., & Berglund, Å. F. (2016). The operator 4.0: Human cyber-physical systems & adaptive automation towards human-automation symbiosis work systems. *IFIP Advances in Information and Communication Technology*, *488*, 677–686. https://doi.org/10.1007/978-3-319-51133-7_80

Romero, D., & Vernadat, F. (2016). Enterprise information systems state of the art: Past, present and future trends. *Computers in Industry*, *79*, 3–13. https://doi.org/10.1016/J.COMPIND.2016.03.001

Romero, L. F., & Arce, A. (2017). Applying Value Stream Mapping in Manufacturing: A Systematic Literature Review. *IFAC-PapersOnLine*, *50*(1), 1075–1086. https://doi.org/10.1016/J.IFACOL.2017.08.385

Rother, Mike., Shook, John., & Lean Enterprise Institute. (2003). *Learning to see : value stream mapping to create value and eliminate muda*. Lean Enterprise Institute.

Rudberg, M., & Wikner, J. (2004). Mass customization in terms of the customer order decoupling point. *Production Planning and Control*, *15*(4), 445–458. https://doi.org/10.1080/0953728042000238764

Russell, N., van der Aalst, W. M. P., & ter Hofstede, A. H. M. (2006a). Exception Handling Patterns in Process-Aware Information Systems. In *BPM Center, Report BPM-06-04*. http://bpmcenter.org/wp-content/uploads/reports/2006/BPM-06-04.pdf

Russell, N., van der Aalst, W., & ter Hofstede, A. (2006b). Workflow Exception Patterns. In E. Dubois & K. Pohl (Eds.), *Advanced Information Systems Engineering: Vol. 4001 LNCS* (pp. 288–302). Springer, Berlin, Heidelberg. https://doi.org/10.1007/11767138_20

Sadiku, M. N. O., Wang, Y., Cui, S., Musa, S. M., & Perry, R. G. (2017). The INDUSTRIAL INTERNET OF THINGS. *International Journal of Advances in Scientific Research and Engineering (IJASRE), ISSN:2454-8006, DOI: 10.31695/IJASRE*, *3*(11), 1–5. https://doi.org/10.7324/IJASRE.2017.32538

Sahu, C. K., Young, C., & Rai, R. (2020). Artificial intelligence (AI) in augmented reality (AR)-assisted manufacturing applications: a review. *Https://Doi.Org/10.1080/00207543.2020.1859636*, *59*(16), 4903–4959. https://doi.org/10.1080/00207543.2020.1859636

Saldanha, T. J. V., Melville, N. P., Ramirez, R., & Richardson, V. J. (2013). Information systems for collaborating versus transacting: Impact on manufacturing plant performance in the presence of

demand volatility. *Journal of Operations Management*, *31*(6), 313–329. https://doi.org/10.1016/J.JOM.2013.07.007
Salunke, S., Weerawardena, J., & McColl-Kennedy, J. R. (2011). Towards a model of dynamic capabilities in innovation-based competitive strategy: Insights from project-oriented service firms. *Industrial Marketing Management*, *40*(8), 1251–1263. https://doi.org/10.1016/j.indmarman.2011.10.009
Salvador, F., Martin de Holan, P., & Piller, F. (2009). Cracking the Code of Mass Customization. *MIT Sloan Management Review*, *50*(3), 71–78.
Sanchez, M., Exposito, E., & Aguilar, J. (2020). Industry 4.0: survey from a system integration perspective. *International Journal of Computer Integrated Manufacturing*, *33*(10–11), 1017–1041. https://doi.org/10.1080/0951192X.2020.1775295
Sawhney, R. (2006). Interplay between uncertainty and flexibility across the value-chain: Towards a transformation model of manufacturing flexibility. *Journal of Operations Management*, *24*(5), 476–493. https://doi.org/10.1016/j.jom.2005.11.008
Scheuerlein, H., Rauchfuss, F., Dittmar, Y., Molle, R., Lehmann, T., Pienkos, N., & Settmacher, U. (2012). New methods for clinical pathways-Business Process Modeling Notation (BPMN) and Tangible Business Process Modeling (t.BPM). *Langenbeck's Archives of Surgery*, *397*(5), 755–761. https://doi.org/10.1007/S00423-012-0914-Z
Schild, K., & Bussmann, S. (2007). Self-organization in manufacturing operations. *Communications of the ACM*, *50*(12), 74–79. https://doi.org/10.1145/1323688.1323698
Schmidbauer, C., Hader, B., & Schlund, S. (2021). Evaluation of a Digital Worker Assistance System to enable Adaptive Task Sharing between Humans and Cobots in Manufacturing. *Procedia CIRP*, *104*, 38–43. https://doi.org/10.1016/J.PROCIR.2021.11.007
Schmidt, D. C. (2006). Model-driven engineering. *Computer*, *39*(2), 25–31. https://doi.org/10.1109/MC.2006.58
Schonberger, D., Lindorfer, R., & Froschauer, R. (2018). Modeling Workflows for Industrial Robots Considering Human-Robot-Collaboration. *Proceedings - IEEE 16th International Conference on Industrial Informatics, INDIN 2018*, 400–405. https://doi.org/10.1109/INDIN.2018.8471999
Schönig, S., Ackermann, L., & Jablonski, S. (2018). *Internet of Things Meets BPM: A Conceptual Integration Framework*. https://doi.org/10.5220/0006824803070314
Schönig, S., Ackermann, L., Jablonski, S., & Ermer, A. (2020). IoT meets BPM: a bidirectional communication architecture for IoT-aware process execution. *Software and Systems Modeling*, *19*(6), 1443–1459. https://doi.org/10.1007/s10270-020-00785-7
Schroeder, C. (2016). The Challenges of Industry 4 . 0 for Small and Medium-sized Enterprises. *Friedrich Ebert Foundation*, *August*, 1–28. www.fes-2017plus.de
Schuh, G. G., Anderl, R., Gausemeier, J. J., ten Hompel, M. M., Wahlster, W. (Eds.), Ander, Lr., Gausemeier, J. J., ten Hompel, M. M., & Wahlster, W. (Eds.). (2020). Industrie 4.0 Maturity Index. Managing the Digital Transformation of Companies. In *Acatech Study*. www.acatech.de/publikationen.
Schulte, S., Hoenisch, P., Hochreiner, C., Dustdar, S., Klusch, M., & Schuller, D. (2014). Towards Process Support for Cloud Manufacturing. *Proceedings . IEEE 18th International Enterprise Distributed Object Computing Conference*, *2014-Decem*(December), 142–149. https://doi.org/10.1109/EDOC.2014.28
Schulte, S., Schuller, D., Steinmetz, R., & Abels, S. (2012). Plug-and-play virtual factories. *IEEE Internet Computing*, *16*(5), 78–82. https://doi.org/10.1109/MIC.2012.114
Seethamraju, R. (2015). Adoption of Software as a Service (SaaS) Enterprise Resource Planning (ERP) Systems in Small and Medium Sized Enterprises (SMEs). *Information Systems Frontiers*, *17*(3), 475–492. https://doi.org/10.1007/S10796-014-9506-5/TABLES/1
Shahrubudin, N., Lee, T. C., & Ramlan, R. (2019). An Overview on 3D Printing Technology: Technological, Materials, and Applications. *Procedia Manufacturing*, *35*, 1286–1296. https://doi.org/10.1016/j.promfg.2019.06.089
Shanmugapriya, P., & Suresh, R. M. (2012). Software Architecture Evaluation Methods-A survey. *International Journal of Computer Applications*, *49*(16), 975–8887.

Shawish, A., & Salama, M. (2014). Cloud Computing: Paradigms and Technologies. *Studies in Computational Intelligence*, *495*, 39–67. https://doi.org/10.1007/978-3-642-35016-0_2
Sjödin, D. R., Parida, V., Leksell, M., & Petrovic, A. (2018). Smart Factory Implementation and Process Innovation. *Https://Doi.Org/10.1080/08956308.2018.1471277*, *61*(5), 22–31. https://doi.org/10.1080/08956308.2018.1471277
Spijkers, D. (2019). *Expressing and supporting buffering and (un)bundling in the manufacturing domain using BPMN 2.0 with a case study in the high tech manufacturing industry* [Master thesis, Eindhoven University of Technology]. https://research.tue.nl/en/studentTheses/expressing-and-supporting-buffering-and-unbundling-in-the-manufac
Stearns, P. N. (2020). *The Industrial Revolution in World History (5th Edition)*. Routledge.
Steinau, S., Andrews, K., & Reichert, M. (2018). The relational process structure. In J. Krogstie & H. Reijers (Eds.), *Advanced Information Systems Engineering. CAiSE 2018: Vol. 10816 LNCS* (pp. 53–67). Springer Verlag. https://doi.org/10.1007/978-3-319-91563-0_4/FIGURES/5
Stentoft, J., Jensen, K. W., Philipsen, K., & Haug, A. (2019). Drivers and Barriers for Industry 4.0 Readiness and Practice: A SME Perspective with Empirical Evidence. *Proceedings of the Annual Hawaii International Conference on System Sciences*, *2019-Janua*, 5155–5164. https://doi.org/10.24251/HICSS.2019.619
Stigler, S., & Oberhauser, R. (2017). Microflows: Enabling Agile Business Process Modeling to Orchestrate Semantically-Annotated Microservices. *Proceedings of the Seventh International Symposium on Business Modeling and Software Design*, 19–28. https://doi.org/10.5220/0006527100190028
Storti, D., & Yurtoglu, M. (2015). *CUDA for engineers : an introduction to high-performance parallel computing*. Addison-Wesley Professional.
Strozzi, F., Colicchia, C., Creazza, A., & Noè, C. (2017). Literature review on the 'Smart Factory' concept using bibliometric tools. *International Journal of Production Research*, *55*(22), 6572–6591. https://doi.org/10.1080/00207543.2017.1326643
Sungur, C. T., Spiess, P., Oertel, N., & Kopp, O. (2013). Extending BPMN for Wireless Sensor Networks. *2013 IEEE 15th Conference on Business Informatics*, 109–116. https://doi.org/10.1109/CBI.2013.24
Swenson, K. D. (2010). *Mastering the unpredictable : how adaptive case management will revolutionize the way that knowledge workers get things done*. Meghan-Kiffer Press.
Swenson, K. D. (2013). Designing for an innovative learning organization. *Proceedings - IEEE International Enterprise Distributed Object Computing Workshop, EDOC*, 209–213. https://doi.org/10.1109/EDOC.2013.38
Szelągowski, M. (2019). BPMN update proposal for non-expert users. *Advances in Intelligent Systems and Computing*, *835*, 681–691. https://doi.org/10.1007/978-3-319-97490-3_64
Tabim, V. M., Ayala, N. F., & Frank, A. G. (2021). Implementing Vertical Integration in the Industry 4.0 Journey: Which Factors Influence the Process of Information Systems Adoption? *Information Systems Frontiers*, *1*, 1–18. https://doi.org/10.1007/S10796-021-10220-X/FIGURES/3
Tan, J., & Wang, L. (2010). Flexibility-efficiency tradeoff and performance implications among Chinese SOEs. *Journal of Business Research*, *63*(4), 356–362. https://doi.org/10.1016/j.jbusres.2009.04.016
Tang, C. S. (2010a). A review of marketing-operations interface models: From co-existence to coordination and collaboration. In *International Journal of Production Economics* (Vol. 125, Issue 1, pp. 22–40). Elsevier. https://doi.org/10.1016/j.ijpe.2010.01.014
Tang, C. S. (2010b). A review of marketing–operations interface models: From co-existence to coordination and collaboration. *International Journal of Production Economics*, *125*(1), 22–40. https://doi.org/10.1016/J.IJPE.2010.01.014
Tang, D., Zheng, K., Zhang, H., Sang, Z., Zhang, Z., Xu, C., Espinosa-Oviedo, J. A., Vargas-Solar, G., & Zechinelli-Martini, J.-L. (2016). Using Autonomous Intelligence to Build a Smart Shop Floor. *Procedia CIRP*, *56*, 354–359. https://doi.org/10.1016/j.procir.2016.10.039

Tao, F., Qi, Q., Wang, L., & Nee, A. Y. C. (2019). Digital Twins and Cyber–Physical Systems toward Smart Manufacturing and Industry 4.0: Correlation and Comparison. *Engineering*, *5*(4), 653–661. https://doi.org/10.1016/J.ENG.2019.01.014

Tedjini, S., Perret, E., Deepu, V., & Bernier, M. (2010). Chipless Tags, the Next RFID Frontier. *The Internet of Things*, 239–249. https://doi.org/10.1007/978-1-4419-1674-7_23

Terkaj, W., Tolio, T., & Urgo, M. (2015). A virtual factory approach for in situ simulation to support production and maintenance planning. *CIRP Annals - Manufacturing Technology*, *64*(1), 451–454. https://doi.org/10.1016/J.CIRP.2015.04.121

Theorin, A., Bengtsson, K., Provost, J., Lieder, M., Johnsson, C., Lundholm, T., & Lennartson, B. (2017). An event-driven manufacturing information system architecture for Industry 4.0. *International Journal of Production Research*, *55*(5), 1297–1311. https://doi.org/10.1080/00207543.2016.1201604

Thoben, K. D., Wiesner, S. A., & Wuest, T. (2017). "Industrie 4.0" and smart manufacturing-a review of research issues and application examples. In *International Journal of Automation Technology* (Vol. 11, Issue 1, pp. 4–16). Fuji Technology Press. https://doi.org/10.20965/ijat.2017.p0004

Tolio, T., Ceglarek, D., Elmaraghy, H. A., Fischer, A., Hu, S. J., Laperrière, L., Newman, S. T., & Váncza, J. (2010). SPECIES—Co-evolution of products, processes and production systems. *CIRP Annals*, *59*(2), 672–693. https://doi.org/10.1016/J.CIRP.2010.05.008

Tomašević, I., Stojanović, D., Slović, D., Simeunović, B., & Jovanović, I. (2021). Lean in High-Mix/Low-Volume industry: a systematic literature review. *Production Planning and Control*, *32*(12), 1004–1019. https://doi.org/10.1080/09537287.2020.1782094

Tönnes, W. (2018). Applying data of historical defects to increase efficiency of rework in assembly. *Procedia CIRP*, *72*, 255–260. https://doi.org/10.1016/J.PROCIR.2018.03.086

Traganos, K., & Grefen, P. (2015). Hybrid service compositions: When BPM meets dynamic case management. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *9306*, 226–239. https://doi.org/10.1007/978-3-319-24072-5_16

Traganos, K., Grefen, P., Vanderfeesten, I., Erasmus, J., Boultadakis, G., & Bouklis, P. (2021). The HORSE framework: A reference architecture for cyber-physical systems in hybrid smart manufacturing. *Journal of Manufacturing Systems*, *61*(November 2020), 461–494. https://doi.org/10.1016/j.jmsy.2021.09.003

Traganos, K., Spijkers, D., Grefen, P., & Vanderfeesten, I. (2020). Dynamic Process Synchronization Using BPMN 2.0 to Support Buffering and (Un)Bundling in Manufacturing. *Lecture Notes in Business Information Processing*, *392 LNBIP*, 18–34. https://doi.org/10.1007/978-3-030-58638-6_2

Traganos, K., Vanderfeesten, I., Grefen, P., Erasmus, J., Gerrits, T., & Verhofstad, W. (2020). End-To-End Production Process Orchestration for Smart Printing Factories: An Application in Industry. *Proceedings - 2020 IEEE 24th International Enterprise Distributed Object Computing Conference, EDOC 2020*, 155–164. https://doi.org/10.1109/EDOC49727.2020.00027

Tran, K. P. (2021). Artificial Intelligence for Smart Manufacturing: Methods and Applications. *Sensors 2021, Vol. 21, Page 5584*, *21*(16), 5584. https://doi.org/10.3390/S21165584

Truijens, J. (1990). *Informatie-Infrastructuur: een Instrument voor het Management*.

Tsarouchi, P., Makris, S., & Chryssolouris, G. (2016). Human – robot interaction review and challenges on task planning and programming. *International Journal of Computer Integrated Manufacturing*, *29*(8), 916–931. https://doi.org/10.1080/0951192X.2015.1130251

Tseng, M. M., & Piller, F. T. (2003). The Customer Centric Enterprise. In M. M. Tseng & F. T. Piller (Eds.), *The Customer Centric Enterprise*. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-55460-5

Um, J., Lyons, A., Lam, H. K. S., Cheng, T. C. E., & Dominguez-Pery, C. (2017). Product variety management and supply chain performance: A capability perspective on their relationships and competitiveness implications. *International Journal of Production Economics*, *187*, 15–26. https://doi.org/10.1016/J.IJPE.2017.02.005

Unver, H. O. (2013). An ISA-95-based manufacturing intelligence system in support of lean initiatives. *International Journal of Advanced Manufacturing Technology*, *65*(5–8), 853–866. https://doi.org/10.1007/s00170-012-4223-z
van der Aalst, W. (2009). Petri Nets. In *Encyclopedia of Database Systems* (pp. 2103–2108). Springer US. https://doi.org/10.1007/978-0-387-39940-9_817
van der Aalst, W. (2012). *Process mining: discovering and improving Spaghetti and Lasagna processes*. 1–7. https://doi.org/10.1109/CIDM.2011.6129461
van der Aalst, W., Artale, A., Montali, M., & Tritini, S. (2017). Object-Centric Behavioral Constraints: Integrating Data and Declarative Process Modelling. *Proceedings of the 30th International Workshop on DescriptionLogics (CEUR Workshop Proceedings; Vol. 1879)*. http://ceur-ws.org/Vol-1879/paper51.pdf
van der Aalst, W. M. P. (1994). Putting high-level Petri nets to work in industry. *Computers in Industry*, *25*(1), 45–54. https://doi.org/10.1016/0166-3615(94)90031-0
van der Aalst, W. M. P. (2013). Business Process Management: A Comprehensive Survey. *ISRN Software Engineering*, *2013*, 1–37. https://doi.org/10.1155/2013/507984
van der Aalst, W. M. P., Barthelmess, P., Ellis, C. A., & Wainer, J. (2001). Proclets: A framework for Lightweight interacting workflow processes. *International Journal of Cooperative Information Systems*, *10*(04), 443–481. https://doi.org/10.1142/S0218843001000412
van der Aalst, W. M. P., ter Hofstede, A. H. M., & Weske, M. (2003). Business Process Management: A Survey. *Lecture Notes in Computer Science*, *2678*, 1–12. https://doi.org/10.1007/3-540-44895-0_1
van der Aalst, W., & van Hee, K. (2002). *Workflow Management Models, Methods, and Systems*.
van Gorp, P., Vanderfeesten, I., Dalinghaus, W., Mengerink, J., van der Sanden, B., & Kubben, P. (2013). Towards generic MDE support for extracting purpose-specific healthcare models from annotated, unstructured texts. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *7789 LNCS*, 213–221. https://doi.org/10.1007/978-3-642-39088-3_14
Vanderfeesten, I., Erasmus, J., Traganos, K., Bouklis, P., Garbi, A., Boultadakis, G., Dijkman, R., & Grefen, P. (2019). Developing Process Execution Support for High-Tech Manufacturing Processes. *Empirical Studies on the Development of Executable Business Processes*, 113–142. https://doi.org/10.1007/978-3-030-17666-2_6
Vargo, S. L., & Lusch, R. F. (2004). The Four Service Marketing Myths Remnants of a Goods-Based, Manufacturing Model. *Journal of Service Research*, *6*(4), 324–335. https://doi.org/10.1177/1094670503262946
Venable, J., Pries-Heje, J., & Baskerville, R. (2017). FEDS: a Framework for Evaluation in Design Science Research. *Https://Doi.Org/10.1057/Ejis.2014.36*, *25*(1), 77–89. https://doi.org/10.1057/EJIS.2014.36
Verdonschot, J. (2020). *Operational exception handling in a smart (discrete) manufacturing production environment* [Master thesis, Eindhoven University of Technology]. https://pure.tue.nl/ws/portalfiles/portal/164236449/Master_Thesis_John_Verdonschot.pdf
Verschuren, P. J. M., & Doorewaard, H. (2010). *Designing a research project* (R. Poper & M. J. Mellion, Eds.; 2nd editio). Eleven International Publishing.
Villani, V., Pini, F., Leali, F., & Secchi, C. (2018). Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications. *Mechatronics*, *55*, 248–266. https://doi.org/10.1016/J.MECHATRONICS.2018.02.009
Vještica, M., Dimitrieski, V., Pisarić, M., Kordić, S., Ristić, S., & Luković, I. (2021). Towards a Formal Specification of Production Processes Suitable for Automatic Execution. *Open Computer Science*, *11*(1), 161–179. https://doi.org/10.1515/COMP-2020-0200/MACHINEREADABLECITATION/RIS
Wan, J., Tang, S., Shu, Z., Li, D., Wang, S., Imran, M., & Vasilakos, A. v. (2016). Software-Defined Industrial Internet of Things in the Context of Industry 4.0. *IEEE Sensors Journal*, *16*(20), 7373–7380. https://doi.org/10.1109/JSEN.2016.2565621

Wang, H.-N., He, Q.-Q., Zhang, Z., Peng, T., & Tang, R.-Z. (2021). Framework of automated value stream mapping for lean production under the Industry 4.0 paradigm *. *J Zhejiang Univ-Sci A (Appl Phys & Eng) 2021*, *22*(5), 382–395. https://doi.org/10.1631/jzus.A2000480
Wang, L., & Su, L. (2011). Implementing green publishing and printing process with workflow management system. *International Conference on Management and Service Science, MASS 2011*. https://doi.org/10.1109/ICMSS.2011.5998573
Wang, L., Törngren, M., & Onori, M. (2015). Current status and advancement of cyber-physical systems in manufacturing. *Journal of Manufacturing Systems*, *37*, 517–527. https://doi.org/10.1016/J.JMSY.2015.04.008
Wang, P., & He, W. (2012). Research on key performance indicator (KPI) of business process. *Proceedings of the 2012 2nd International Conference on Business Computing and Global Informatization, BCGIN 2012*, 151–154. https://doi.org/10.1109/BCGIN.2012.46
Wang, S., Wan, J., Li, D., & Zhang, C. (2016). Implementing Smart Factory of Industrie 4.0: An Outlook. *International Journal of Distributed Sensor Networks*, *12*(1), 3159805. https://doi.org/10.1155/2016/3159805
Wang, S., Yang, S., & Cai, H. (2018). Application Research of Workflow Technology in Printing and Publishing Industry. *Proceedings of the 2018 VII International Conference on Network, Communication and Computing - ICNCC 2018*. https://doi.org/10.1145/3301326
Wang, T., Thomas, D. J., & Rudi, N. (2014). The Effect of Competition on the Efficient-Responsive Choice. *Production and Operations Management*, *23*(5), 829–846. https://doi.org/10.1111/poms.12093
Wang, Y., Ma, H. S., Yang, J. H., & Wang, K. S. (2017). Industry 4.0: a way from mass customization to mass personalization production. *Advances in Manufacturing*, *5*(4), 311–320. https://doi.org/10.1007/s40436-017-0204-7
Wang, Y., Wang, S., Yang, B., Gao, B., & Wang, S. (2020). An effective adaptive adjustment method for service composition exception handling in cloud manufacturing. *Journal of Intelligent Manufacturing*, *33*(3), 735–751. https://doi.org/10.1007/S10845-020-01652-4/TABLES/10
Weske, M. (2012). Business process management: Concepts, languages, architectures, second edition. *Business Process Management: Concepts, Languages, Architectures, Second Edition*, 1–403. https://doi.org/10.1007/978-3-642-28616-2/COVER
West, M. (2011). Developing High Quality Data Models. In *Developing High Quality Data Models* (Version 2). https://doi.org/10.1016/c2009-0-30508-5
Weyer, S., Schmitt, M., Ohmer, M., & Gorecky, D. (2015). Towards industry 4.0 - Standardization as the crucial challenge for highly modular, multi-vendor production systems. *IFAC-PapersOnLine*, *28*(3), 579–584. https://doi.org/10.1016/j.ifacol.2015.06.143
Wieringa, R. (2009). Design Science as Nested Problem Solving. *Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology - DESRIST '09*. https://doi.org/10.1145/1555619
Williams, T. J. (1990). A Reference Model for Computer Integrated Manufacturing from the Viewpoint of Industrial Automation. *IFAC Proceedings Volumes*, *23*(8), 281–291. https://doi.org/10.1016/S1474-6670(17)51748-6
Witsch, M., & Vogel-Heuser, B. (2012). Towards a formal specification framework for manufacturing execution systems. *IEEE Transactions on Industrial Informatics*, *8*(2), 311–320. https://doi.org/10.1109/TII.2012.2186585
Wodtke, D., & Weikum, G. (1997). A formal foundation for distributed workflow execution based on state charts. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *1186*, 230–246. https://doi.org/10.1007/3-540-62222-5_48
Wohed, P., Aalst, W. M. P. van der, Dumas, M., Hofstede, A. H. M. ter, & Russell, N. (2006). On the Suitability of BPMN for Business Process Modelling. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *4102 LNCS*, 161–176. https://doi.org/10.1007/11841760_12

Wohlin, C. (2014). Guidelines for Snowballing in Systematic Literature Studies and a Replication in Software Engineering. *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering - EASE '14*. https://doi.org/10.1145/2601248

Woodward, J. (1965). *Industrial organization : theory and practice*. Oxford University Press.

Wortmann, F., & Flüchter, K. (2015). Internet of Things: Technology and Value Added. *Business and Information Systems Engineering*, *57*(3), 221–224. https://doi.org/10.1007/S12599-015-0383-3/FIGURES/2

Wu, D., Greer, M. J., Rosen, D. W., & Schaefer, D. (2013). Cloud manufacturing: Drivers, current status, and future trends. *ASME 2013 International Manufacturing Science and Engineering Conference Collocated with the 41st North American Manufacturing Research Conference, MSEC 2013*, *2*, MSEC2013-1106. https://doi.org/10.1115/MSEC2013-1106

Wübbeke, J., Meissner, M., Zenglein, M. J., & Ives, J. (2016). The making of a high-tech superpower and consequences for industrial countries. In *MERICS PAPERS* (Issue 2).

Xu, X. (2012). From cloud computing to cloud manufacturing. *Robotics and Computer-Integrated Manufacturing*, *28*(1), 75–86. https://doi.org/10.1016/J.RCIM.2011.07.002

Yannakakis, M. (2000). Hierarchical State Machines. In J. van Leeuwen, O. Watanabe, M. Hagiya, P. D. Mosses, & T. Ito (Eds.), *Theoretical Computer Science: Exploring New Frontiers of Theoretical Informatics. TCS 2000: Vol. 1872 LNCS* (pp. 315–330). Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-44929-9_24

Yao, X., Zhang, J., Li, Y., & Zhang, C. (2018). Towards flexible rfid event-driven integrated manufacturing for make-to-order production. *International Journal of Computer Integrated Manufacturing*, *31*(3), 228–242. https://doi.org/10.1080/0951192X.2017.1407455

Yousfi, A., Bauer, C., Saidi, R., & Dey, A. K. (2016). uBPMN. *Information and Software Technology*, *74*, 55–68. https://doi.org/10.1016/J.INFSOF.2016.02.002

Zhang, L., Luo, Y., Tao, F., Li, B. H., Ren, L., Zhang, X., Guo, H., Cheng, Y., Hu, A., & Liu, Y. (2014). Cloud manufacturing: a new manufacturing paradigm. *Enterprise Information Systems*, *8*(2), 167–187. https://doi.org/10.1080/17517575.2012.683812

Zhong, R. Y., Xu, X., Klotz, E., & Newman, S. T. (2017). Intelligent Manufacturing in the Context of Industry 4.0: A Review. *Engineering*, *3*(5), 616–630. https://doi.org/10.1016/J.ENG.2017.05.015

Zhou, F., Lin, X., Liu, C., Zhao, Y., Xu, P., Ren, L., Xue, T., & Ren, L. (2019). A survey of visualization for smart manufacturing. *Journal of Visualization*, *22*(2), 419–435. https://doi.org/10.1007/S12650-018-0530-2/FIGURES/7

Zhu, J., & Li, Y. L. (2012). A framework for printing on demand based on workflow. *Proceedings - 2012 5th International Conference on Intelligent Computation Technology and Automation, ICICTA 2012*, 14–17. https://doi.org/10.1109/ICICTA.2012.11

Zhu, L., Johnsson, C., Mejvik, J., Varisco, M., & Schiraldi, M. (2018). Key performance indicators for manufacturing operations management in the process industry. *IEEE International Conference on Industrial Engineering and Engineering Management*, *2017-December*, 969–973. https://doi.org/10.1109/IEEM.2017.8290036

Zimniewicz, M. (2020). *SHOP4CF Architecture. Deliverabe D3.2. SHOP4CF Project*. https://live-shop4cf.pantheonsite.io/wp-content/uploads/2021/07/SHOP4CF-WP3-D32-DEL-210119-v1.0.pdf

Zissis, D., & Lekkas, D. (2012). Addressing cloud computing security issues. *Future Generation Computer Systems*, *28*(3), 583–592. https://doi.org/10.1016/J.FUTURE.2010.12.006

Zor, S., Görlach, K., & Leymann, F. (2010). Using BPMN for Modeling Manufacturing Processes. In *Conference on Manufacturing* (pp. 1–8). http://www.iaas.uni-stuttgart.de/RUS-data/INPROC-2010-48 - Using BPMN for Modeling Manufacturing Processes.pdf

Zor, S., Schumm, D., & Leymann, F. (2011). A Proposal of BPMN Extensions for the Manufacturing Domain. *Proceedings of the 44th CIRP International Conference on Manufacturing Systems*.

Appendix A – Terms and Abbreviations

Table 30 provides definitions for the terms that are prominent in this thesis. Table 31 clarifies used abbreviations.

Table 30: Definitions of terms that are core in this thesis.

Term	Definition	Source
Business process	The combination of a set of activities within an enterprise with a structure describing their logical order and dependence whose objective is to produce a desired result	(Aguilar-Savén, 2004)
Business Process Management	A paradigm for supporting business processes using methods, techniques, and software to design, enact, control, and analyze operational processes involving humans, organizations, applications, documents and other sources of information	(van der Aalst et al., 2003)
Complex system/Process complexity	Refers to a system containing uncertainty during the development process or intrinsically in its design, the outcome not being fully predictable or controlled.	(Elmaraghy et al., 2012)
Exception	Any event that disrupts the normal behavior of the designed manufacturing operations.	Various
Industry 4.0	Industry 4.0 is a term originally coined by the German Academy of Science and Engineering (acatech) to describe the fourth stage of industrialization, in a national initiative to secure the future of the German manufacturing industry.	(Kagermann et al., 2013)
Manufacturing	Refers to the application of physical and chemical processes to alter the geometry, properties, and/or appearance of a given starting material to make parts or products; it also includes assembly of multiple parts to make products	(Groover, 2010)
Manufacturing process	A set of activities (subprocess or tasks) in a combination of machinery, tools, power, and labor that transforms inputs (materials, product specifications, production order) into outputs (product or assembly).	(Groover, 2010; International Standards Organization, 2015)
Manufacturing System	A collection of integrated equipment and human resources that performs one or more processing and/or assembly operations on a starting work material, part, or set of parts.	(Groover, 2010)
Smart factory	a smart factory embraces and integrates the recent Industry 4.0 technological advances in computer networks, data integration and analytics to bring transparency to manufacturing units	(Lee, 2015)
Smart manufacturing	Used to characterize the current, new traits of manufacturing in the ongoing fourth industrial revolution, both from business and technology perspectives.	Various

Table 31: Abbreviations appearing in this thesis.

Abbreviation	Meaning
AGV	Automated Guided Vehicle
API	Application Programming Interface
AR	Augmented Reality
ATO	Assemble to Order
BAS	Batch Automation System
BI	Business Intelligence
BOS	BOSCH (SHOP4CF project pilot)
BPA	Business Process Analytics
BPI	Business Process Intelligence
BPM	Business Process Management
BPMN	Business Process Model & Notation
BPMS	Business Process Management System
BPMS-RA	Business Process Management System-Reference Architecture
BTS	Build to Stock
CAM	Computer Aided Manufacturing
CEP	Complex Event Processing
CIM	Computer Integrated Manufacturing
CMMN	Case Management Model & Notation
CMMS	Computerized Maintenance Management System
CNC	Computer Numerically Controlled
CPP	Canon Production Printing (OEDIPUS project pilot)
CPPS	Cyber-physical Production System
CPS	Cyber-physical system
CRM	Customer Relationship Management
DCM	Dynamic Case Management
DCS	Distributed Control System
DML	Dedicated Manufacturing Line
DMN	Decision Model & Notation
DMS	Dedicated Machining System
DSR	Design Science Research
DSRM	Design Science Research Methodology
DTO	Design to Order
E2E	End-to-End
EIS	Enterprise Information Systems
EPC	Event-driven Process Chain
ERP	Enterprise Resource Planning
ETO	Engineer to Order
EU	European Union
FEDS	Framework for Evaluation in Design Science
FIFO	First In First Out
FMS	Flexible Manufacturing System
FTY	First Time Yield
HCI	Human Computer Interface
HCS	High Capacity Stacker
HMI	Human Machine Interface
HRC	Human Robot Collaboration
HRTM	Human Robot Time & Motion
IaaS	Infrastructure as a Service
IDEF	Integration DEfinition
IEC	International Electrotechnical Commission
IF	Interface
INCOSE	International Council on Systems Engineering

IoT	Internet-of-Things
IIoT	Industrial Internet-of-Things
ItU	Intention to Use
IS	Information System
JDF	Job Definition Format
JMF	Job Messaging Format
JSON	JavaScript Object Notation
K4+1	Kruchten 4+1 views framework
KPIs	Key Performance Indicators
LIFO	Last In First Out
M2M	Machine-to-Machine
MD	Model Driven
MDA	Model Driven Architecture
MDE	Model Driven Engineering
MDD	Model Driven Development
MDSE	Model Driven Software Engineering
MES	Manufacturing Executions System
MESA	Manufacturing Enterprise Solutions Association
MIS	Management Information System
MOM	Manufacturing Operations Management
MPC	Manufacturing Planning & Control
MPM	Manufacturing Process Management
MPMS	Manufacturing Process Management System
MTO	Make to Order
MTS	Make to Stock
NIST	National Institute Standards and Technology
OEE	Overall Equipment Effectiveness
OMG	Object Management Group
OPC-UA	Open Platform Communications-Unified Architecture
OSGi	Open Services Gateway initiative
PaaS	Platform as a Service
PCM	Production Case Management
PEoU	Perceived Ease of Use
PMS	Production Management System
PPMS	Printing Process Management System
PLC	Programmable Logical Controller
PLM	Product Lifecycle Management
PU	Perceived Usefulness
QMS	Quality Management System
RA	Reference Architecture
RAMI4.0	Reference Architecture Model for Industry 4.0
REST	Representational State Transfer
RMS	Reconfigurable Manufacturing System
RMT	Reconfigurable Machine Tools
ROS	Robot Operating System
RTM	Robot Time & Motion
SaaS	Software as a Service
SCADA	Supervisory Control and Data Acquisition
SCM	Supply Chain Management
SFC	Sequential Functional Chart
SMEs	Small & Medium-sized Enterprises
SLR	Systematic Literature Review
SOA	Service-Oriented Architecture
TAM	Technology Acceptance Model
TRI	Thomas Regout International (HORSE project pilot)

UI	User Interface
UML	Unified Modeling Language
UT5	Updated Truijens 5-aspect framework
VSM	Value Stream Mapping
WfMC	Workflow Management Coalition
WFMS	Workflow Management System
WIP	Work in Progress
WMS	Warehouse Management System
XJDF	Exchange Job Definition Format

Appendix B – Overview of real-world pilot cases

Table 32 briefly presents the HORSE initial three pilot cases and the seven open call experiments and their projects. More detailed information can be found on http://horse-project.eu/Pilots and http://horse-project.eu/Experiments. Table 33summarizes the EIT OEDIPUS pilot case. Table 34 briefly presents the SHOP4CF initial three pilot cases. More detailed information can be found on https://www.shop4cf.eu/use-cases/.

Table 32: HORSE project pilots and open call experiments

	Acronym	Short description	Manufact. company (end-user)	Sector	Challenges	Solution
HP1	BOS-H	Robotic based quality (visual) inspection and human-robot co-manipulation	Robert Bosch España, Fábrica de Castellet, Barcelona, Spain	Automotive	• Product quality with required cycle time • Customized packaging of wiper systems • Operator's health conditions	• Automated packaging of wiper systems, including artificial visual quality check of parts, to replace current situation (manual) • Augmented Reality assistance for manually checking points on potentially assessed faulty parts • Orchestration and monitoring of robotic and human tasks, including mobile messages to workers
HP2	TRI	Flexible assembly with mobile robot	Thomas Regout International BV, Maastricht, Netherlands	Industrial equipment	• Overall process management of shop floor • Replacing pick & place units by a smart	• Orchestration, monitoring and planning support to enable flexible and effective production • Augmented Reality assistance

	Acronym	Short description	Manufact. company (end-user)	Sector	Challenges	Solution
					robot without fences • Reduce high dependency on experience on tool preparation by introducing Augmented Reality	on production tools assembly
HP3	OPSA	Robot-human hybrid position/force control co-working applications	Odlewnie Polskie SA, Starachowice, Poland	Iron casting	• Automation of the casts cutting process • Improved working conditions and reduce injury risk	• Learning by demonstration for new castings • Automated cutting of metal castings to replace current (manual) situation • Ensure safety and comfort of the workers
HE1	Guided Safety	Guided safety design, configuration and execution of small part assembly process with collaborative robots	Denso Automotive Deutschland GmbH – Production Engineering, Germany	Automotive	• Small batches, highly customized products that require flexibility • Safe design and configuration of work-stations	• Workflow automation • Safety designer • Shift Manager
HE2	Flex Coating	Collaborative Robotics for Industrial Coating Cells	FLUPOL	Coating applications	• Coating done by experienced human operators	• Programming by demonstration • Object recognition and localization

	Acronym	Short description	Manufact. company (end-user)	Sector	Challenges	Solution
HE3	ENDORSE	Effective Robotic GriNDing of Surface Areas through HORSE framework	Enikon Aerospace d.o.o.	Aircraft manufact.	• Automation of grinding process • Automated quality inspection	• Impedance-based robot control algorithm to improve the quality of surface treatment • Automated quality inspection system
HE4	COMPLEMANT	COllaborative robot aMPLifying and Extending huMAN capabiliTies	Ghepi Srl	Injection molding (plastic materials)	• Monitoring operator's working conditions	• Use of cobot to support human operators • Monitoring of operators' parameters • Intervention system to alleviate operators
HE5	BEAUTY	BEnding Automation TYcoon	TETRA Industriservice Group	Metal equipment	• Fully production process automation • Introduce state of art automation tools at a low cost	• End-to-end process management • Programming of the robot for the transportation and manipulation tasks • Integration of modified legacy machines
HE6	ARCO	Autonomous Robot CO-worker	Tintas Robbialac SA	Paints	• Use of an autonomous robot co-worker to carry the materials in warehouse	• Moving robot • Process automation by information system
HE7	RANCH	Robotics And Neural networks Combined in HORSE	Ophardt Belgien	Industrial and medical equipment	• (De)racking and quality inspection of shrouds • Stressing work for	• Robot to derack shrouds • Camera for visual inspection • Neural network application for defect detection

	Acronym	Short description	Manufact. company (end-user)	Sector	Challenges	Solution
					human operators	

Table 33: EIT OEDIPUS project pilot

	Acronym	Short description	Manufact. company (end-user)	Sector	Challenges	Solution
EO1	CPP	Automated handling and transportation of printed media	Canon Production Printing (former Océ), Venlo, Netherlands	Industrial printing	• Transportation of media across multiple stations • Transparency of activities Alleviate human operator's work load with regard to ergonomics	• Process orchestration of transportation activities • Process monitoring for activities overview • Robotic arm for automated media unloading • Autonomous mobile robot with mounted gripper for automated media (un)loading and transportation

Table 34: SHOP4CF project pilots

	Acronym	Short description	Manufact. company (end-user)	Sector	Challenges	Solution
SP1	BOS-S	Efficient multiple-line material loading	Robert Bosch España, Fábrica Madrid S.A., Madrid, Spain	Automotive	• Similar, parallel automated production lines with random utilization rate, based on orders and required type of products • Sharing of agent for	• Multiple-line material loading by shared autonomous mobile robot • Efficient allocation of loading tasks to autonomous shared robot and human operators • Safe autonomous mobile robot navigation in area

	Acronym	Short description	Manufact. company (end-user)	Sector	Challenges	Solution
					loading the lines per need • Minimize utilization of human operators whose main task is at another production areas	shared with human operators
SP2	SAG	Robot training for parts sorting	Siemens AG, Munich, Germany	Industrial equipment	• Precise position of contact pads for automated assembly of electrical switches. • Frequent changes on specifications of contacts pads that require long times on new teaching	• Specialized glove for robot teaching • Modeling of activities for human-robot collaboration
SP3	VWP	Optimized maintenance of skids during cataphoretic electro-coating (KTL) process	Volkswagen, Wrzesnia factory, Poznan, Poland	Automotive	• Disturbances on the conductive properties of skids, used to guide the car bodies during painting process	• Optimized identification of disturbances • Correct and timely management of disturbances • More complete information to operators for maintenance

Appendix C – Manufacturing process model fragments represented in BPMN

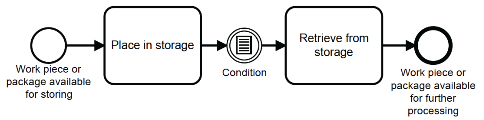
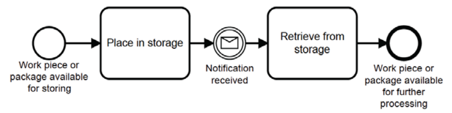
This appendix illustrates a set of manufacturing process model fragments, modeled in BPMN. The set is part of a catalog for representing manufacturing operations in BPMN (Erasmus, Vanderfeesten, Traganos, & Grefen, 2020) and is under concern in Section 3.5 for designing a synchronization mechanism to provide execution support when modeling those.

The process model fragments, representing operations of the four main categories of manufacturing operations (Section 2.1.3), are grouped into two main manufacturing constructs (due to their similar characteristics), namely buffering and bundling/unbundling.

C.1 Process model fragments under buffering category

Two process model fragments fall under the buffering manufacturing construct, listed in Table 35.

Table 35: Process model fragments under buffering manufacturing construct.

Inventory operations	
Buffering	Preservation
	

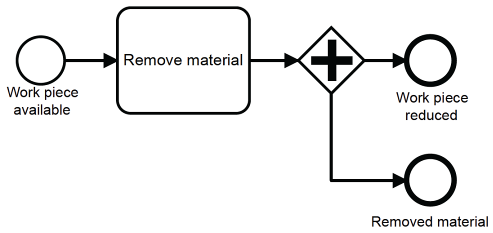
C.2 Process model fragments under bundling/unbundling category

Eight process model fragments fall under the bundling/unbundling manufacturing construct(s), listed in Table 36.

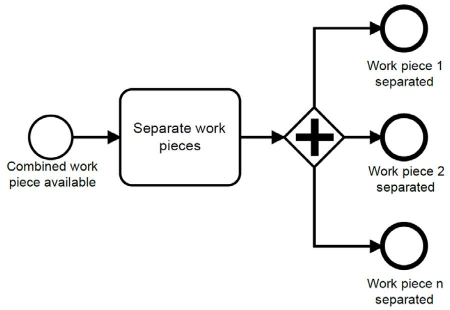
Table 36: Process model fragments under bundling/unbundling manufacturing construct(s).

Production operations

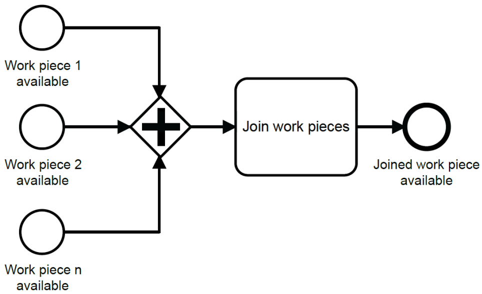
Material removal



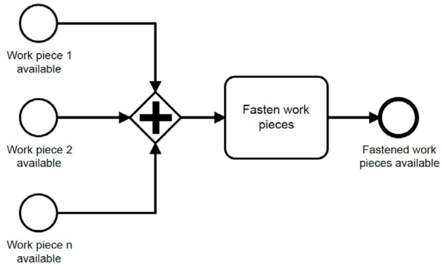
Separating



Permanent joing

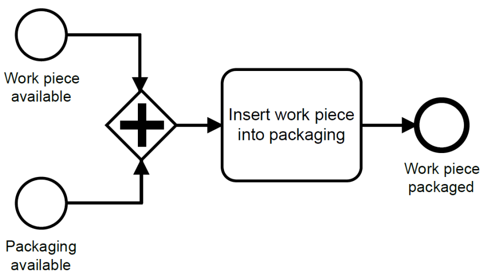


Mechanical fastening

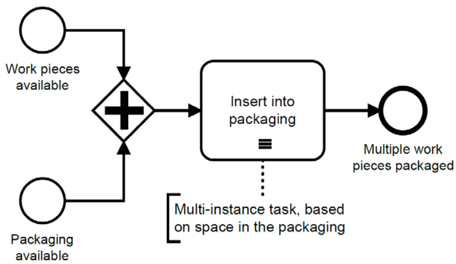


Inventory operations

Individual packaging



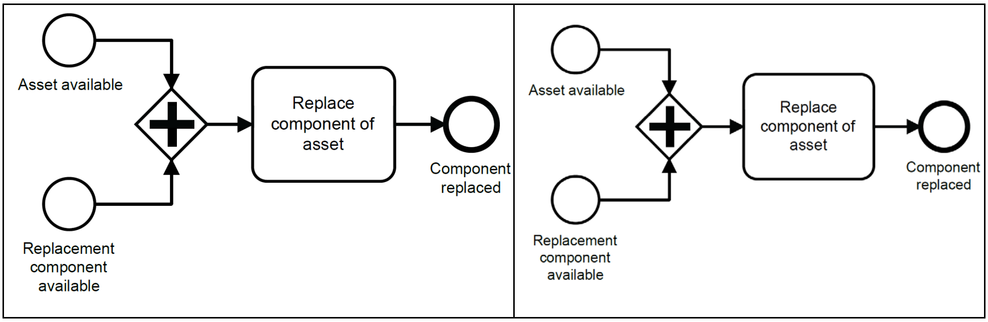
Unitizing



Maintenance operations

Replacing

Scheduled replacing


Asset available
Replacement component available
Replace component of asset
Component replaced
Asset available
Replacement component available
Replace component of asset
Component replaced

Appendix D – Systematic literature review for Ch. 4 – Exception Handling

This appendix presents the details of the Systematic Literature Review (SLR) conducted for Chapter 4 – Exception Handling (Section 4.2.1). Section D.1 presents the search steps and Section D.2 lists the "short list" of the selected studies for data extraction and synthesis.

D.1 Search steps

Table 37 presents the steps of searching of studies for "exception handling" topic. The final search term on Scopus yielded 994 studies. On Science Direct, as the search capabilities are more limited, individual search queries had to be executed. Then aggregated results were checked for duplicates, finally yielding 736 studies.

Table 37: SLR search steps and results for retrieving relevant studies.

Search engine	Search term	Reasoning	Results
Scopus	(TITLE-ABS-KEY (((exception*) W/3 (type OR categor* OR classification OR taxonomy OR pattern OR handling)))) AND (LIMIT-TO (OA , "all")) AND (LIMIT-TO (PUBYEAR , 2021) OR LIMIT-TO (PUBYEAR , 2020) OR LIMIT-TO (PUBYEAR , 2019) OR LIMIT-TO (PUBYEAR , 2018) OR LIMIT-TO (PUBYEAR , 2017) OR LIMIT-TO (PUBYEAR , 2016) OR LIMIT-TO (PUBYEAR , 2015) OR LIMIT-TO (PUBYEAR , 2014) OR LIMIT-TO (PUBYEAR , 2013) OR LIMIT-TO (PUBYEAR , 2012) OR LIMIT-TO (PUBYEAR , 2011) OR LIMIT-TO (PUBYEAR , 2010) OR LIMIT-TO (PUBYEAR , 2009) OR LIMIT-TO (PUBYEAR , 2008) OR LIMIT-TO (PUBYEAR , 2007) OR LIMIT-TO (PUBYEAR , 2006)) AND (LIMIT-TO (LANGUAGE , "English"))	Exploratory search term including only the term "exception" (no synonyms or relevant alternatives).	1656
	(TITLE-ABS-KEY (((exception* OR failure OR error OR deviat* OR defect) W/3 (type OR categor* OR classification OR taxonomy OR pattern OR handling)))) AND (LIMIT-TO (OA , "all")) AND (LIMIT-TO (PUBYEAR , 2021) OR LIMIT-TO (PUBYEAR , 2020) OR LIMIT-TO (PUBYEAR , 2019) OR LIMIT-TO (PUBYEAR , 2018) OR LIMIT-TO (PUBYEAR , 2017) OR LIMIT-TO (PUBYEAR , 2016) OR LIMIT-TO (PUBYEAR , 2015) OR LIMIT-TO (PUBYEAR , 2014) OR LIMIT-TO (PUBYEAR , 2013) OR LIMIT-TO (PUBYEAR , 2012) OR LIMIT-TO (PUBYEAR , 2011) OR LIMIT-TO (PUBYEAR , 2010) OR LIMIT-TO (PUBYEAR , 2009) OR	Including other terms (i.e., failure, error, deviation, defect) that often appear in literature and practice to represent "disturbances" from on objective.	28709

	LIMIT-TO (PUBYEAR , 2008) OR LIMIT-TO (PUBYEAR , 2007) OR LIMIT-TO (PUBYEAR , 2006)) AND (LIMIT-TO (LANGUAGE , "English"))		
	(TITLE-ABS-KEY (((exception* OR failure OR error OR deviat* OR defect) W/3 (type OR categor* OR classification OR taxonomy OR pattern OR handling)))) AND (LIMIT-TO (OA , "all")) AND (LIMIT-TO (PUBYEAR , 2021) OR LIMIT-TO (PUBYEAR , 2020) OR LIMIT-TO (PUBYEAR , 2019) OR LIMIT-TO (PUBYEAR , 2018) OR LIMIT-TO (PUBYEAR , 2017) OR LIMIT-TO (PUBYEAR , 2016) OR LIMIT-TO (PUBYEAR , 2015) OR LIMIT-TO (PUBYEAR , 2014) OR LIMIT-TO (PUBYEAR , 2013) OR LIMIT-TO (PUBYEAR , 2012) OR LIMIT-TO (PUBYEAR , 2011) OR LIMIT-TO (PUBYEAR , 2010) OR LIMIT-TO (PUBYEAR , 2009) OR LIMIT-TO (PUBYEAR , 2008) OR LIMIT-TO (PUBYEAR , 2007) OR LIMIT-TO (PUBYEAR , 2006)) AND (LIMIT-TO (LANGUAGE , "English")) AND (LIMIT-TO (SUBJAREA , "COMP") OR LIMIT-TO (SUBJAREA , "ENGI") OR LIMIT-TO (SUBJAREA , "MATE") OR LIMIT-TO (SUBJAREA , "MULT") OR LIMIT-TO (SUBJAREA , "ENER") OR LIMIT-TO (SUBJAREA , "DECI"))	Limiting results on relevant to this research subject areas (e.g., Engineering) to exclude studies in irrelevant domains (e.g., psychology).	11389
	(TITLE-ABS-KEY (((exception* OR failure OR error OR deviat* OR defect) W/3 (type OR categor* OR classification OR taxonomy OR pattern OR handling)) AND NOT (medic* OR *coding OR code OR programming OR neural OR optical OR training))) AND (LIMIT-TO (OA , "all")) AND (LIMIT-TO (SUBJAREA , "COMP") OR LIMIT-TO (SUBJAREA , "ENGI") OR LIMIT-TO (SUBJAREA , "MATE") OR LIMIT-TO (SUBJAREA , "MULT") OR LIMIT-TO (SUBJAREA , "ENER") OR LIMIT-TO (SUBJAREA , "DECI")) AND (LIMIT-TO (PUBYEAR , 2021) OR LIMIT-TO (PUBYEAR , 2020) OR LIMIT-TO (PUBYEAR , 2019) OR LIMIT-TO (PUBYEAR , 2018) OR LIMIT-TO (PUBYEAR , 2017) OR LIMIT-TO (PUBYEAR , 2016) OR LIMIT-TO (PUBYEAR , 2015) OR LIMIT-TO (PUBYEAR , 2014) OR LIMIT-TO (PUBYEAR , 2013) OR LIMIT-TO (PUBYEAR , 2012) OR LIMIT-TO (PUBYEAR , 2011) OR LIMIT-TO (PUBYEAR , 2010) OR LIMIT-TO (PUBYEAR , 2009) OR LIMIT-TO (PUBYEAR , 2008) OR LIMIT-TO (PUBYEAR , 2007) OR LIMIT-TO (PUBYEAR , 2006)) AND (LIMIT-TO (LANGUAGE , "English"))	Excluding specific terms that further limit down irrelevant studies. For instance, from a quick scanning on titles of the search results, medical studies were still appearing, despite not having include “Medical” or “Healthcare” as subject areas in the previous step. Or, also, studies on “abnormalities” on image processing should be excluded.	8164
	(TITLE-ABS-KEY (((exception* OR failure OR error OR deviat* OR defect) W/3 (type OR categor* OR classification OR taxonomy OR pattern OR handling)) AND (process OR *flow) AND NOT (medic* OR *coding OR code OR programming OR neural OR optical OR training))) AND (LIMIT-TO (OA , "all")) AND (LIMIT-TO (SUBJAREA , "COMP") OR LIMIT-TO (SUBJAREA , "ENGI") OR LIMIT-TO (SUBJAREA , "MATE") OR LIMIT-TO (SUBJAREA , "MULT") OR LIMIT-TO (SUBJAREA , "ENER") OR LIMIT-TO (SUBJAREA , "DECI")) AND (LIMIT-TO (PUBYEAR , 2021) OR LIMIT-TO (PUBYEAR , 2020) OR LIMIT-TO (PUBYEAR , 2019) OR LIMIT-TO (PUBYEAR , 2018) OR LIMIT-TO (PUBYEAR , 2017) OR LIMIT-TO (PUBYEAR , 2016) OR LIMIT-TO (PUBYEAR , 2015) OR LIMIT-TO (	Scoping results towards “process” or “workflow” domain.	1994

	PUBYEAR , 2014) OR LIMIT-TO (PUBYEAR , 2013) OR LIMIT-TO (PUBYEAR , 2012) OR LIMIT-TO (PUBYEAR , 2011) OR LIMIT-TO (PUBYEAR , 2010) OR LIMIT-TO (PUBYEAR , 2009) OR LIMIT-TO (PUBYEAR , 2008) OR LIMIT-TO (PUBYEAR , 2007) OR LIMIT-TO (PUBYEAR , 2006)) AND (LIMIT-TO (LANGUAGE , "English"))		
	(TITLE-ABS-KEY (((exception* OR failure OR error OR deviat* OR defect) W/3 (type OR categor* OR classification OR taxonomy OR pattern OR handling)) AND (manufacturing OR process OR *flow) AND NOT (medic* OR *coding OR code OR programming OR neural OR optical OR training))) AND (LIMIT-TO (OA , "all")) AND (LIMIT-TO (SUBJAREA , "COMP") OR LIMIT-TO (SUBJAREA , "ENGI") OR LIMIT-TO (SUBJAREA , "MATE") OR LIMIT-TO (SUBJAREA , "MULT") OR LIMIT-TO (SUBJAREA , "ENER") OR LIMIT-TO (SUBJAREA , "DECI")) AND (LIMIT-TO (PUBYEAR , 2021) OR LIMIT-TO (PUBYEAR , 2020) OR LIMIT-TO (PUBYEAR , 2019) OR LIMIT-TO (PUBYEAR , 2018) OR LIMIT-TO (PUBYEAR , 2017) OR LIMIT-TO (PUBYEAR , 2016) OR LIMIT-TO (PUBYEAR , 2015) OR LIMIT-TO (PUBYEAR , 2014) OR LIMIT-TO (PUBYEAR , 2013) OR LIMIT-TO (PUBYEAR , 2012) OR LIMIT-TO (PUBYEAR , 2011) OR LIMIT-TO (PUBYEAR , 2010) OR LIMIT-TO (PUBYEAR , 2009) OR LIMIT-TO (PUBYEAR , 2008) OR LIMIT-TO (PUBYEAR , 2007) OR LIMIT-TO (PUBYEAR , 2006)) AND (LIMIT-TO (LANGUAGE , "English"))	Adding also "manufacturing" domain for more specific scoping.	2066
	(TITLE-ABS-KEY (((exception* OR failure OR error OR deviat*) W/3 (type OR categor* OR classification OR taxonomy OR pattern OR handling)) AND (manufacturing OR process OR *flow) AND NOT (medic* OR *coding OR code OR programming OR neural OR optical OR training))) AND (LIMIT-TO (OA , "all")) AND (LIMIT-TO (SUBJAREA , "COMP") OR LIMIT-TO (SUBJAREA , "ENGI") OR LIMIT-TO (SUBJAREA , "MATE") OR LIMIT-TO (SUBJAREA , "MULT") OR LIMIT-TO (SUBJAREA , "ENER") OR LIMIT-TO (SUBJAREA , "DECI")) AND (LIMIT-TO (PUBYEAR , 2021) OR LIMIT-TO (PUBYEAR , 2020) OR LIMIT-TO (PUBYEAR , 2019) OR LIMIT-TO (PUBYEAR , 2018) OR LIMIT-TO (PUBYEAR , 2017) OR LIMIT-TO (PUBYEAR , 2016) OR LIMIT-TO (PUBYEAR , 2015) OR LIMIT-TO (PUBYEAR , 2014) OR LIMIT-TO (PUBYEAR , 2013) OR LIMIT-TO (PUBYEAR , 2012) OR LIMIT-TO (PUBYEAR , 2011) OR LIMIT-TO (PUBYEAR , 2010) OR LIMIT-TO (PUBYEAR , 2009) OR LIMIT-TO (PUBYEAR , 2008) OR LIMIT-TO (PUBYEAR , 2007) OR LIMIT-TO (PUBYEAR , 2006)) AND (LIMIT-TO (LANGUAGE , "English"))	From a quick scanning, the term "defect" resulted in studies that have a narrow scope (e.g., on product quality) than the current research. So, the term "defect*" was removed. Given that the rest terms provide a lot of coverage, this exclusion should not have great impact on rigor.	1494
	(TITLE-ABS-KEY (((exception* OR failure OR error OR deviat*) W/3 (type OR categor* OR classification OR taxonomy OR pattern OR handling)) AND (manufacturing OR process OR *flow) AND NOT (medic* OR *coding OR code OR programming OR software OR neural OR optical OR training OR simulat* OR {type I	Excluding studies focusing on simulation or mathematical problems.	994

	error} OR {type II error}))) AND (LIMIT-TO (OA , "all")) AND (LIMIT-TO (SUBJAREA , "COMP") OR LIMIT-TO (SUBJAREA , "ENGI") OR LIMIT-TO (SUBJAREA , "MATE") OR LIMIT-TO (SUBJAREA , "MULT") OR LIMIT-TO (SUBJAREA , "ENER") OR LIMIT-TO (SUBJAREA , "DECI")) AND (LIMIT-TO (PUBYEAR , 2021) OR LIMIT-TO (PUBYEAR , 2020) OR LIMIT-TO (PUBYEAR , 2019) OR LIMIT-TO (PUBYEAR , 2018) OR LIMIT-TO (PUBYEAR , 2017) OR LIMIT-TO (PUBYEAR , 2016) OR LIMIT-TO (PUBYEAR , 2015) OR LIMIT-TO (PUBYEAR , 2014) OR LIMIT-TO (PUBYEAR , 2013) OR LIMIT-TO (PUBYEAR , 2012) OR LIMIT-TO (PUBYEAR , 2011) OR LIMIT-TO (PUBYEAR , 2010) OR LIMIT-TO (PUBYEAR , 2009) OR LIMIT-TO (PUBYEAR , 2008) OR LIMIT-TO (PUBYEAR , 2007) OR LIMIT-TO (PUBYEAR , 2006)) AND (LIMIT-TO (LANGUAGE , "English"))	The result of 994 studies was deemed to be a good base for analysis, so that was the final search term on Scopus.	
ScienceDirect	((Exception types) OR (Exception Categorization) OR (Exception Category) OR (Exception Classification) OR (Exception Taxonomy) OR (Exception Pattern) OR (Exception Handling)) AND (Workflow OR Process) Search on: Title, abstract or author-specified keywords Filters: Open Access	Focusing on combination of terms for “exception” term. Scoping into process or workflow domain studies.	14
	((Exception types) OR (Exception Categorization) OR (Exception Category) OR (Exception Classification) OR (Exception Taxonomy) OR (Exception Pattern) OR (Exception Handling)) AND (Manufacturing) Search on: Title, abstract or author-specified keywords Filters: Open Access	Focusing on combination of terms for “exception” term. Scoping into manufacturing domain.	6
	((Failure types) OR (Failure Categorization) OR (Failure Category) OR (Failure Classification) OR (Failure Taxonomy) OR (Failure Pattern) OR (Failure Handling)) AND (Workflow OR Process) Search on: Title, abstract or author-specified keywords Filters: Open Access / SubjectArea IN (Engineering)	Focusing on combination of terms for “failure” term. Scoping into process or workflow domain studies.	365
	((Failure types) OR (Failure Categorization) OR (Failure Category) OR (Failure Classification) OR (Failure Taxonomy) OR (Failure Pattern) OR (Failure Handling)) AND (Manufacturing) Search on: Title, abstract or author-specified keywords Filters: Open Access / SubjectArea IN (Engineering)	Focusing on combination of terms for “failure” term. Scoping into manufacturing domain.	116
	((Error types) OR (Error Categorization) OR (Error Category) OR (Error Classification) OR (Error Taxonomy) OR (Error Pattern) OR (Error Handling)) AND (Workflow or Process) Search on: Title, abstract or author-specified keywords	Focusing on combination of terms for “error” term. Scoping into process or workflow domain studies.	54

	Filters: Open Access / SubjectArea IN (Engineering, Decision Sciences)		
	((Error types) OR (Error Categorization) OR (Error Category) OR (Error Classification) OR (Error Taxonomy) OR (Error Pattern) OR (Error Handling)) AND (Manufacturing) Search on: Title, abstract or author-specified keywords Filters: Open Access / SubjectArea IN (Engineering, Decision Sciences)	Focusing on combination of terms for "error" term. Scoping into manufacturing domain.	15
	((Deviation types) OR (Deviation Categorization) OR (Deviation Category) OR (Deviation Classification) OR (Deviation Taxonomy) OR (Deviation Pattern) OR (Deviation Handling)) AND (Workflow OR Process) Search on: Title, abstract or author-specified keywords Filters: Open Access / SubjectArea IN (Engineering, Decision Sciences)	Focusing on combination of terms for "deviation" term. Scoping into process or workflow domain studies.	138
	((Deviation types) OR (Deviation Categorization) OR (Deviation Category) OR (Deviation Classification) OR (Deviation Taxonomy) OR (Deviation Pattern) OR (Deviation Handling)) AND (Manufacturing) Search on: Title, abstract or author-specified keywords Filters: Open Access / SubjectArea IN (Engineering, Decision Sciences)	Focusing on combination of terms for "deviation" term. Scoping into manufacturing domain.	48

D.2 Short list of selected studies

Table 38 lists the final list of selected studies for analysis for "exception handling".

Table 38: "Short list" of selected studies for "exception handling"

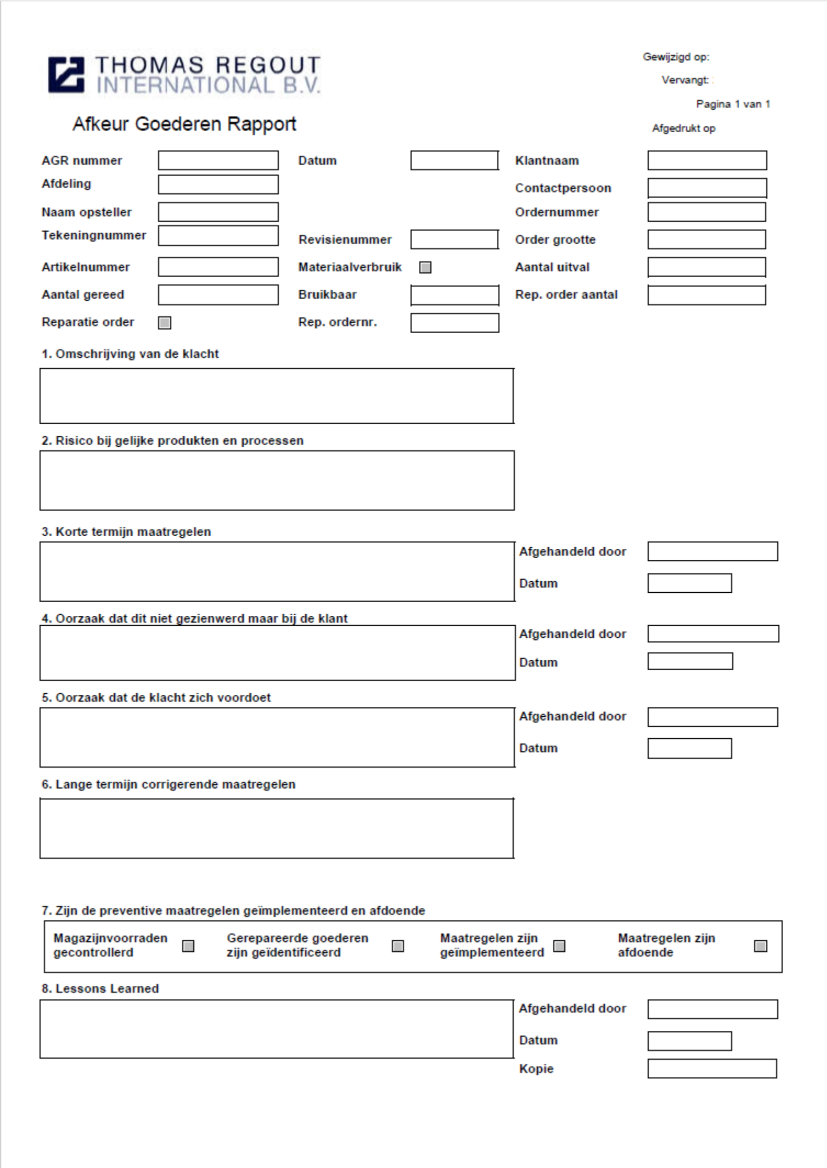
No	Author(s) (Date)	Title
1	Chavez et al. (2021)	Industry 4.0, transition or addition in SMEs? A systematic literature review on digitalization for deviation management
2	Dahl et al. (2021)	Application of the Sequence Planner Control Framework to an Intelligent Automation System with a Focus on Error Handling
3	Klamann & Winner (2021)	Comparing Different Levels of Technical Systems for a Modular Safety Approval—Why the State of the Art Does Not Dispense with System Tests Yet
4	Psarommatis & Kiritsis (2021)	A hybrid Decision Support System for automating decision making in the event of defects in the era of Zero Defect Manufacturing
5	Riedel et al. (2021)	A deep learning-based worker assistance system for error prevention: Case study in a real-world manual assembly
6	Andree et al. (2020)	Exception handling in the context of fragment-based case management
7	Chavez et al. (2020)	Digital Tools and Information Needs Assessment for Efficient Deviation Handling in SMEs
8	Filz et al. (2020)	Simulation-based Assessment of Quality Inspection Strategies on Manufacturing Systems
9	Hossayni et al. (2020)	SemKoRe: Improving Machine Maintenance in Industrial IoT with Semantic Knowledge Graphs
10	Islam et al. (2020)	The implementation of preventive maintenance using machine damage analysis: a case study of power plant machine
11	Kim et al. (2020)	Errors in Human-Robot Interactions and Their Effects on Robot Learning
12	Muqimuddin & Singgih (2020)	Integrated FMEA-MCDM for Prioritizing Operational Disruption in Production Process
13	Qian et al. (2020)	ResourceNet: a collaboration network among decentralised manufacturing resources for autonomous exception-handling in smart manufacturing
14	Xiong et al. (2020)	Detecting data flow errors across processes in business process collaboration
15	Cao et al. (2019)	An Ontology-based Approach for Failure Classification in Predictive Maintenance Using Fuzzy C-means and SWRL Rules
16	Franciosi et al. (2019)	A taxonomy of performance shaping factors for human reliability analysis in industrial maintenance
17	Lukens et al. (2019)	Best Practices Framework for Improving Maintenance Data Quality to Enable Asset Performance Analytics
18	Müller et al. (2019)	Situational cognitive assistance system in rework area
19	Sexton et al. (2019)	Categorization Errors for Data Entry in Maintenance Work-Orders
20	Calvo Olivares et al. (2018)	A novel qualitative prospective methodology to assess human error during accident sequences
21	Tomiyama & Moyen (2018)	Resilient architecture for cyber-physical production systems
22	Tönnes (2018)	Applying data of historical defects to increase efficiency of rework in assembly
23	Athamena & Houhamdi (2017)	An Exception Management Model in Multi-Agents Systems
24	Zhang et al. (2017)	Agent and cyber-physical system based self-organizing and self-adaptive intelligent shopfloor
25	Akkaya et al. (2016)	Systems Engineering for Industrial Cyber-Physical Systems Using Aspects
26	Böllhoff et al. (2016)	Evaluation of the Human Error Probability in Cellular Manufacturing
27	Caputo et al. (2016)	Selection of assembly lines feeding policies based on parts features
28	Farooqui et al. (2016)	Error handling within highly automated automotive industry: Current practice and research needs
29	Keddis et al. (2016)	Handling errors in dynamic production environments

30	Ritter & Sosulski (2016)	Exception Handling in Message-Based Integration Systems and Modeling Using BPMN
31	Caputo et al. (2015)	Modeling Errors in Kitting Processes for Assembly Lines Feeding
32	Kassner & Mitschang (2015)	MaXCept - Decision support in exception handling through unstructured data integration in the production context: An integral part of the smart factory
33	Kujawińska & Vogt (2015)	Human factors in visual quality control
34	Mok et al. (2015)	Determination of Failure Cause in Remanufacturing
35	Reyes et al. (2015)	Association between Human Error and Occupational Accidents' Contributing Factors for Hand Injuries in the Automotive Manufacturing Industry
36	Sahno et al. (2015)	Framework for continuous improvement of production processes
37	Stich et al. (2015)	Big Data Technology for Resilient Failure Management in Production Systems
38	Bauer et al. (2014)	Concept of a Failures Management Assistance System for the Reaction on Unforeseeable Events during the Ramp-up
39	Leitner & Rinderle-Ma (2014)	A systematic review on security in Process-Aware Information Systems – Constitution, challenges, and future directions
40	Ritter & Sosulski (2014)	Modeling Exception Flows in Integration Systems
41	Schuh et al. (2014)	Methodology for the Evaluation of Forecast Reliability of Production Planning Systems
42	Adam & Aurich (2013)	Classification of Seemingly Random Failures Using Similarity Analysis
43	Depaire et al. (2013)	A process deviation analysis framework
44	Faria & Azevedo (2013)	On the Reliability Evaluation of Failure Delayed Industrial Systems
45	Srewil & Scherer (2013	Effective Construction Process Monitoring and Control through a Collaborative Cyber-Physical Approach
46	Stavenko et al. (2013)	Process Model Reasoning: From Workflow to Case Management
47	Ali & Reiff-Marganiec (2012)	Autonomous failure-handling mechanism for WF long running transactions
48	Ferreira Da Silva et al. (2012)	Self-healing of operational workflow incidents on distributed computing infrastructures
49	Reichert & Weber (2012)	Exception Handling. In Enabling Flexibility in Process-Aware Information Systems
50	Ruan et al. (2012)	A user-defined exception handling framework in the VIEW scientific workflow management system
51	Antunes (2011)	BPM and exception handling: Focus on organizational resilience
52	Derbali (2011)	A Framework Proposal for Intelligent Management of Unexpected Exceptions in Workflow
53	Gao & Xu (2010)	An intelligent agent-assisted logistics exception management decision support system: A design science approach
54	Lerner et al. (2010)	Exception handling patterns for process modeling
55	Misic et al. (2010)	Concept of the exception handling system for manufacturing business processes
56	de Snoo et al. (2010)	Coordination activities of human planners during rescheduling: case analysis and event handling procedure
57	Wang et al. (2009)	Constraint integration and violation handling for BPEL processes
58	Weidlich et al. (2009)	Vertical Alignment of Process Models – How Can We Get There?
59	Wu (2009)	A new method of exception handling in workflow
60	Balakrishnan et al. (2008)	A strategic framework for managing failure in jit supply chains
61	Gaaloul et al. (2008)	A secure task delegation model for workflows
62	Adams et al. (2007)	Dynamic, extensible and context-aware exception handling for workflows
63	Bruccoleri et al. (2007)	Reconfiguration: a key to handle exceptions and performance deteriorations in manufacturing operations
64	Hamadi et al. (2007)	Self-adapting recovery nets for policy-driven exception handling in business processes
65	Leymann & Roller (2006)	Modeling business processes with BPEL4WS

66	Russell et al. (2006a)	Exception Handling Patterns in Process-Aware Information Systems
67	Russell et al. (2006b)	Workflow Exception Patterns
68	Wang & Wang (2006)	From process logic to business logic—A cognitive approach to business process management
69	Adams et al. (2005)	Facilitating flexibility and dynamic exception handling in workflows through worklets
70	Mourão & Antunes (2005)	A collaborative framework for unexpected exception handling
71	Saastamoinen (2005)	Exception-Based Approach for Information Systems Evaluation: The Method and its Benefits to Information Systems Management
72	Luo et al. (2000)	Exception Handling in Workflow Systems
73	Casati (1999)	A discussion on approaches to handling exceptions in workflows
74	Casati et al. (1999)	Specification and Implementation of Exceptions in Workflow Management Systems
75	Eder & Liebhart (1995)	The Workflow Activity Model (WAMO)
76	Strong & Miller (1995)	Exceptions and exception handling in computerized information processes

Appendix E – Exceptions "Reject goods" report of TRI

Figure 167 shows the report called "Reject goods" at HORSE TRI pilot case, used to briefly describe the occurred exceptions (used to analyze data exceptions from industry (Section 4.2.2.1)).



THOMAS REGOUT INTERNATIONAL B.V.

Gewijzigd op:

Vervangt:

Pagina 1 van 1

Afgedrukt op

Afkeur Goederen Rapport

AGR nummer		Datum		Klantnaam	
Afdeling				Contactpersoon	
Naam opsteller				Ordernummer	
Tekeningnummer		Revisienummer		Order grootte	
Artikelnummer		Materiaalverbruik	☐	Aantal uitval	
Aantal gereed		Bruikbaar		Rep. order aantal	
Reparatie order	☐	Rep. ordernr.			

1. Omschrijving van de klacht

2. Risico bij gelijke produkten en processen

3. Korte termijn maatregelen

Afgehandeld door

Datum

4. Oorzaak dat dit niet gezienwerd maar bij de klant

Afgehandeld door

Datum

5. Oorzaak dat de klacht zich voordoet

Afgehandeld door

Datum

6. Lange termijn corrigerende maatregelen

7. Zijn de preventive maatregelen geïmplementeerd en afdoende

Magazijnvoorraden gecontrollerd ☐ Gerepareerde goederen zijn geïdentificeerd ☐ Maatregelen zijn geïmplementeerd ☐ Maatregelen zijn afdoende ☐

8. Lessons Learned

Afgehandeld door

Datum

Kopie

Figure 167: "Reject goods" report at HORSE TRI pilot case.

Appendix F – Semi-structured interviews for Ch. 4 – Exception Handling

This appendix presents the details of the semi-structured interviews conducted for Chapter 4 – Exception Handling, and more specifically the input on exceptions from industry (Section 4.2.2.2). Section F.1 shows the questionnaire used as a basis for the interviews. Section F.2 presents the responses to the questionnaire by a SHOP4CF pilot, which was not available for an interview, but could respond on paper. Section F.3 and Section F.4 present respectively the transcripts of interviews with two other SHOP4CF pilots.

F.1 Questionnaire for semi-structured interviews

Machine-related exceptions

1) How often (on average) does one of the machines, or a part within a machine, break down during a process?
   a) Would you categorize this exception type as critical or non-critical?
   b) How do you keep a process running when a machine becomes unusable due to breakdown?

Order-related exceptions

2) How much freedom do you offer to customers to change details of their order (such as the amount or delivery date) after they place an order?
   a) How does an order alteration influence the manufacturing process, if at all?
3) What percentage of orders that you receive ultimately get canceled?
   a) How do canceled orders influence the manufacturing process, if at all?
4) Do you offer the possibility to customers to place a rush order? (an order with a shorter than usual delivery time)?
   a) If yes, what percentage of your orders represent rush orders?
      i) How do these rush orders affect the manufacturing process and the delivery time of regular orders?
5) Do you have a formal way to alter the production plan when order-related exceptions occur?
   a) If yes, what do you do if there is not enough time or resources to formally alter the production plan?

Process-related exceptions

6) Have you ever had errors caused by missing parts in the input of a machine or a manufacturing station? (for instance, when parts are fed into a machine by trays, does any issue occur in the process when the tray is not completely filled?)
   a) If yes: How often (on average) does such exception occur? Would you categorize this exception type as critical or non-critical?
      i) Literature suggests that, when such an exception occurs, the production is halted and continues when the missing parts are available. Is this also the case for you, or do you use another approach?
7) What type of systems/software do you have in place to control and monitor manufacturing processes?
   a) Have you ever encountered any software bugs in these systems?

i) If yes: How often (on average) does such exception occur? Would you categorize this exception type as critical or non-critical?
   (1) According to literature, software bugs are solved by making changes to the system/software itself. Is this also the case for you, or do you use another approach?

b) Have you ever encountered any exceptions caused by activity failures within these systems (a failure occurring when the system tries to execute a task/step/activity)?
   i) If yes: How often (on average) does such exception occur? Would you categorize this exception type as critical or non-critical?
      (1) If an exception occurs due to activity failures, an approach is to retry it (the activity); Is this also your approach?
         (a) If retrying an activity does not work, what is your approach to solve an activity failure?

8) Do you have a system in place that can stop manufacturing processes (either within a workcell, production line(s) or production area(s)) in case it detects an impending error?
   a) If yes: How often (on average) is this system activated? And if it is activated, does it cause a significant time loss in production (in other words, how much time it takes to bring the production back to normal)?
9) Is it possible for operations to occur in a wrong sequence? (for instance, a part going through two machines/stations in the wrong order)
   a) If yes: How often (on average) does such exception occur? Would you categorize this exception type as critical or non-critical?
      i) How do you solve this type of exception?

Data-related exceptions

10) How often does an error occur in the data flow between your information systems and/or machines?
    a) How do such errors influence the manufacturing process?
    b) Would you categorize this exception type as critical or non-critical?
11) How often has an exception been caused by a message flow failure? (a function of a system which involves sending/receiving data)
12) How often has an exception been caused by a timeout? (when a message takes too long to be processed)
13) How do you cope with these data flow related exceptions?

Various exceptions

14) Are there any other exceptions/deviations/errors that you can think of that might occur or that have already occurred that have not been covered by the previous questions?
    a) If yes: What type of exceptions are these?
       i) How often (on average) do these exceptions occur? Would you categorize these exceptions as critical or non-critical?
       ii) How did you solve these exceptions?

F.2 Response from SHOP4CF pilot A (on paper)

Machine-related exceptions

1) Rarely, we have the maintenance plan in place, and this reduces the possibility of unknown outages.

a) Critical.
b) Usually, we have redundant lines or buffers to manage these issues but this means we will have some delays due to overproduction.

Order-related exceptions

2) Not known, our ordering is placed by the ERP system but sometimes we do prioritize some orders.
   a) Order prioritization does not change that much the whole manufacturing process.
3) Low, below 10%.
   a) We get new orders from ERP.
4) Not known, our ordering is placed by the ERP system but sometimes we do prioritize some orders.
   a) Low percentage, below 20%.
      i) Not that much.
5) We do prioritize orders if the resources and time are available.
   a) N/A.

Process-related exceptions

6) If the system is not fed properly, the machine will cause an error and most likely it stops or does not even start. However, it depends on the machine and the machinery it is around, e.g., for a manually fed machine the error can be overwritten due to an operator command, but if it is completely automatic it stops and wait for further information.
   a) If it is a fully automatic machine, it should be categorized as critical, if it is manually fed it should be non-critical. In average this happens 2 times a week.
      i) As literature.
7) Usually, it is logged in the MES, however information to the MES is sent via PLC connected devices (e.g., proximity sensor, visual sensor) with some logic behind.
   a) Rarely if a reliable detection method is used.
      i) If the monitoring is not well performed it should be critical.
         (1) Sometimes also sensors are interchanged.
   b) Not really.
      i) N/A.
         (1) If it happens, retry method is applied.
            (a) The engineering team or system integrator is called for support.
8) Yes, safety mechanisms in case errors occur are present. They usually stop one part of the factory as long we have buffers for other machines.
   a) Rarely, but if it happens, it can lead to significant loss of time (compared to power outages which are solved with power outages saving mechanisms, e.g., UPS).
9) No, this is really unlikely, if it happens is due to operator error.
   a) Really unlikely.
      i) Usually through monitoring we prevent such errors happening.

Data-related exceptions

10) Rarely, if it happens it is due to human error. To prevent this, we usually have a locked DB which no operator can directly communicate with.
   a) Inconsistencies in the whole system can occur.
   b) Really critical.
11) Rarely, the main reason are connectivity issues.
12) Rarely, the main reason are connectivity issues.

13) The exceptions are monitored and classified. In case of self-maintained systems, a root cause analysis is performed. In case of external serviced systems, a service is scheduled.

Various exceptions

14) Yes
    a) Human-related errors.
        i) Often. It depends on the machine, sometimes critical and sometimes non-critical.
        ii) We apply a continuous improvement process. If an error occurs, the process/system/interface/training is improved to prevent it in the future.

F.3 Transcript of interview with SHOP4CF pilot B

Two research members conducted the interview and two practitioners participated. To ease the transcription of the interview, the former are presented as [Ri] and the latter as [Pi].

[R1] So, yeah, I'll go through each of the main categories I've gained from my exception classification and just ask what types of exceptions occur.
[P1] Could you share your screen?
[R1] Yeah, I can share my interview questions.
[R1] All right. Have you ever had any errors caused by missing parts that you input in the machine?
[P1] We are producing cars, and this is a big problem because you cannot paint an incomplete car in the normal way. When you have the paint shop and the body shop, must send to you complete car. If something happens it's possible to paint again as some part which is demontable. When you have some doors, sliding doors or something, you can paint again. If something happens in the end of the process. Yeah, but in the normal way you have a big problem and you shouldn't paint car body which they are not complete.
[R1] So this isn't possible to occur?
[P1] To good quality. It's possible. But for the good quality and for the good rules of the production, you must produce paint or car body. Because, when on the beginning, we have some treatment and some preparation. And this is very important for our surface should be painted in one process. On one layer.
[P2] On the other hand, maybe also some problems with the paint also.
[P1] Yeah, paint quality.
[R1] Moving on then to the next question. I assume that you have some kind of system or software monitoring the manufacturing process, what type of system is this?
[P1] We have a lot of systems, one of the most important is the system ERP, this is the enterprise resource planning, but this is for the whole and for the long term systems, but directly for the daily production, we have the manufacturing execution system and these systems help us to look for the production for each station, and this system gives us information about some errors from the counters, how many cars we produce and so on. Later on the end of this meeting, I show you our manufacturing execution systems, how they looks and what kind of information I can see.
[R1] And, have you ever encountered any software bugs in these systems or
[P1] On the beginning we have some problems with databases because I don't know why we have one big databases or a lot of information for our free department from the body shop like paint shop and assembly shop. We have one database. And one year ago, we built a

standalone database, each database for each of our smaller departments, for the body shop, assembly, shop and paint shop. And now we have better data quality and faster response. And we have a big problem with our lift because our data... or maybe not data, but connection between systems and the handshake and talking between these two. A lot of system was very long and we have a problem with the good connection to our lift, which gives us from our storage, some car body, which it should put directly to line, we needed a short time, and to this time it's too big and we have a lot of gaps, from this result. And we have a big problem and this problem solved, some men from India and they are from Microsoft and they must go to us and change something.
[R1] All right.
[P1] Well, this is from the firma their problem, and the firma they pay for this
[R1] So these bugs were mostly solved by outside companies
[P1] Outside company, because this is this happened on the beginning, we just had the production of the implementation phase, and we have the firma take care and stay one year with us and solve our problem. This is in the contract. And they help us learn how to make the system better, solve software bugs.
[R1] Yeah. So outside of the implementation phase, you didn't encounter any more bugs.
[P1] Maybe we have smaller but something, different like a normal production must happen, and it this happen sometimes one or, first time for five years. Some signals are missing or something, but this is a very strange situation. Something happened which should not happen. In the normal production of these two systems works OK, and something happened and they don't work together OK, and our mass system now is crazy because they have the wrong information from this smaller software.
[R2] And maybe, sorry Gijs, if I go back to Jurek's question, so, I think this is a good example. So if we have a manufacturing process, whatever a process and such error occurs and maybe then in MPMS it should have a way to trigger, I don't know, to create a ticket for the external company to solve this bug. So this kind of solution we look for.
[P1] It should prepare Ticket and should prepare some logs. Some tickets is not enough. Should be prepared some logs of information on what happened, what signals they have timestamp and so on.
[R2] Exactly. And yeah, we are looking at the ways that once you identify these errors within the process to automatically create this creation of tickets, etcetera.
[R1] All right. So, still keeping with those monitoring systems, have you ever encountered an exception caused by an activity failure within the system, so when those systems try to execute a task or step in the process or an activity, and that fails. Have you ever encountered that, or would you say that it's the same as a software bug maybe?
[P1] I must translate some. Encountered, what that means.
[P1] Now we have some software bugs, like, they go from the... but this is not software bug. Sometimes we have, or last winter, we have problem with our hardware because one of the signals don't implement from the correct input and now our system doesn't work correctly. This is not maybe not software bug but this is typical hardware bug. But it happens when we have the minus 20 degrees in the winter and we should something close and this are not close, and we have a problem with our chiller, because this chiller are too cold and they was damaged. Yeah, well, not software bugs, but this is the bugs, typical from the hardware and from the wrong electrical implementation. Now for our firms, which they are building for us. But this happened last winter, four years from the start of production. And what happened because we have a very strong winter, minus 20 degrees and the maybe not correct answer for your question, but I want to talk or show you what kind of problem now we have yeah?

[R1] All right. Moving on. If a power outage were to occur, how would this affect your manufacturing process?
[P1] Power outage. When we have big problem with the power we have, because we have a lot of power. We have some power which either go for our machine from the robots and from the PC's, they have power. And we don't have a problem because we have frozen all data. But now we have a problem with the movement because we have the different type of power, the high voltage for your access to the servo motors. And now we have a problem with the, ventilation, like, ventilation of the conveyor, and with the motor of the robots. And some parts they have a buffer and some do not have. And we have to solve this problem because in the direct part of our factory, because we have the current from two different places. One from the {inaudible} and one from the {inaudible}. And we solve this problem from the double, from both the sources.. But, if something happens when we have some peaks, we feel on our production because sometimes something is stopped and now we know what happened. Of course, our colleagues from the 'werktechnik', this is the special department, which they can give us the power and the heating and the cold water, they give us information as something happened. But you feel on the production lights go down and so on. But when you have the good system and good, that means when you have some buffer like additional components which help you frozen your data and you don't have problem with this. It may be happened one time. But like I said, this is some small problems. Much smaller, maybe half hour, and you can run maybe sometimes faster, but, many years ago when I worked in Poznan, something burning in our supply station and we stay more like eight hours. Like one shift.
[R1] All right. Do you have a system in place that can shut down the manufacturing process or maybe only part of it when it detects that an error might occur soon, or not?
[P1] Hmm. Like I said, we have 'werktechnik' in German, I don't know in English, main station, which I prepare for our smaller departments. Cold water, power supply and heating. And if something happened in this area... and gas because we need gas for our truck there and, if something happen in this area, we stop. But, manufacturing process, this is about medium, but in the manufacturing process, we have the additional parts, which we implement to car. And now like we have in this time, some problem with our semiconductors and we stop production. But this is through the two machines in the wrong order and all of our cars, we have a lot of data in our cars. If something happened on the reading point or when you have the production line, if something happen and you must put the car to the different way in manual, from the hand from the manual control, sometimes something happens with your data and this data shift for the wrong car body and you have car body, which they are big, and second one, they are small. And when you shift this control manual, you can put this data from this big car to the small. And in this way it's a problem, but probably you shouldn't have a problem with your station because you see this is big problem and they stop or they want to open something what not exist. But when you shift from the smaller car body to bigger and the system, because the robots are blind, and this car body can't go to the line, system thinking oh, this is a small car body and this is big, you have damage. And then you should have had a system which to confirm your data.
[P2] I don't know if I understand this question correctly, but to me the question is whether it is possible for example, stop the printing process to take out this car or take out the skid and continue something like that.
[P1] We have serial production. When you have a problem with the car body, you must wait for your colleagues, which give you the information. We can paint this car again or we can go to the second run here. But if something happened directly on this line, on the station, you can paint twice, but not good for quality. but is possible.

[R1] In your process, you use machines, I assume. How often does one of those machines or part within one of those machines break down?
[P1] We have a lot of robots and robots don't break down so often. But more problems we have with equipment because we have a lot rotary parts and sometimes we have problem with reading the correct value of the speed of this rotary turbine, or we have a problem with the quality of the paint. But on the machine we must focus more time we have a problem with the equipment, not with the robots. We can exception type of critical, not critical. We have warning and error. If something happened and you don't have the correct value, you have a warning. If something, has bigger value than, maybe 20000 RPM, you have error.
[P2] I don't know whether problems with the skids and pendulums also count in here.
[R1] Um, yeah. I think we can take those in this category.
[P2] But still, they are not so frequent, if I understand correctly.
[R2] And what does this mean. Is it like one once a month, once a week, once a day? How often those things happen?
[P1] on per week, two per week, but we have in the standard three per shift skids go to our maintenance department to check them. And sometimes we have additional problem with our skids because they are a change to dimension or something. And this is one or two per week. This is very stable process, but sometimes happen, because the skids often change temperature. From the room temperature of 20 degrees and goes to almost 200. You have this process: go to paint, go to current and they are painted and you must go to a different area with them and the skid they are cleaning in the hot temperature for the aura with the chemistry, especially chemical input to the skids. And they have a lot of possibility to changing something with the shape.
[R1] All right, now moving on to order related exceptions, so I assume you supply the cars you produce to somewhere, let's call them customers.
[P1] Maybe, because like I said, we have the montage and we have the body shop. if something
happened in the body shop, but this car body goes to us and in this moment, customer from the body shop says to us, look, we have a problem with this car body because we check with our logs and we want to go back with this car body to our body shop to welding something. It's possible with just some places to put this car body to the..., maybe not trollies, but from the fork lift. And with the fork lift, they go back to the body shop, they correct something, welding some parts, or something to check with some quality department and go work for us. That is possible to go back with the process. Yeah, it's possible. It's not so often, but it's possible that something happens.
[R2] Well, I think Gijs here with order related exceptions, he means with the business orders, like I am a customer and I want a red Volkswagen Polo and I don't know, last minute I changed that. I want to black one or something like this.

[P1] Yeah. In this case, yes, it's possible, but not when the car body was produced, because the this is the long term production preparate, and you don't have enough time to provide some equipment part. When you order, stay and wait for the production, impossible to change. But when they go to your production, go to your RPS system, which they are planning for your equipment, because you have a lot of cable, a lot of plastics and with colours and so on. And it is not possible.
[R1] So, if I understand correctly, when you start producing, it's impossible. All right.
[P1] Because we must collect your data about your cars and you must look: this or that, I have this and all this, I have this equipment to cover, to make a big mix and prepare your production there with the line. You have a lot of exception how should look your production

or how long time car body spent in each area and so on. And special, maybe seventy percent of our cars, they are white. But we have a lot of the special colors and we must prepare this car and this part. And sometimes you must wait for your pink car body a little longer. But I'm thinking from the side of the paint shop. But when you want to buy maybe your Polo, in the harlequin color with big wheels and a special navigation system, you must wait maybe a long, long, long time.
[R2] What is the harlequin colour?
[P1] You don't see the harlequin? This is each of the part of the car body are different color.
[R2] Ah, I see.
[P3] This is in the normal production, you can maybe not now, but a long time ago you can buy this car.
[R2] But Marcin is it possible for me if I were ordering a car to kind of make you do it faster, like to pay more or, I don't know, do something and and then you will produce the car faster for me than for other people? Or is it even possible to make you make that car faster or not?
[P3] Everything is possible. when you have the connection you can have this car a little faster, but you must drink a lot of vodka with your friends and you can produce a faster car. But in the normal way, it depends on the situation, on how many cars do you have or which colors and so on. And sometimes it's faster by car, which they are not. Maybe you're 100 percent like you say, like you want, but it's producing now and wait for you. But in the official way, it's probably hard to produce a little faster, but maybe wouldn't be faster and everything is OK. We produce cardboard and it might be. Three days, yeah one day from one of the departments. This is all process of the preparation. You must wait for your places to production. Maybe a few weeks. Because we must collect all this data, prepare to our supplier information on what type, how many plastics, how many wheels, how many navigation system we need and so on. And to some the direct process of the production is not so long, maybe three days, but this is all preparation, all this time to prepare at this car to send it to our dealers. This is the long time. And we have some rules and each car body should stay in place, which they are. Something happened in the one maybe one month of production we have better possibility to replace damaged part because, you know, if something happened in your supplier, they don't now put these parts to you, you implemented it in the car and go with this car to dealer. You buy this car and say, we don't know what happened with your car because in our in our situation, this car is for our e-commerce. This car body should run every time. And in this case, when something happened, we with this part we have a little longer time to replace it and nothing happens. Not special way to please our clients, go to our dealers. We must replace something. We must give you the different car to your business. Now we have the short time when we know where is the car, which place, which parking. And this is the main reason how long time you must wait for your car. So some process is maybe three days. But we have some rules for preparation and to put this car to dealers, to the different country.
[R1] And is it possible for orders to get cancelled once you've started producing them, or not?
[P3] Probably is possible to cancel, but you must pay some contract-euro probably when you cancel. It's possible, but it depends on the situation. We're talking about the car body, which are prepared for our business. And this is not unique. We produce probably a lot of the cars look the same. It's not a problem. But when you produce your Lamborghini than this is big problem with the green leather or something. Everything is possible, but you lost your money and we are prepared for the next client. But it depends on the situation in which area they are, the car body.
[R1] Yeah, so if you haven't started producing the car yet, then it's less problematic if the order gets cancelled. All right.

[R2] Gijs, if I may suggest, given the time, maybe it's more useful if, Martin presents a bit the process and what he finds, because I think more or less the data flow related exceptions are a bit covered, can be bugs or can be machine, let's say malfunctions. So I think we can yeah. Speed up a bit and get more insight from Marcin, from the process.
[R1] Sounds good. Then I'll stop sharing my screen.

F.4 Transcript of interview with SHOP4CF pilot C

Two research members conducted the interview and three practitioners participated. To ease the transcription of the interview, the former are presented as [Ri] and the latter as [Pi].

[R1] Have you ever had errors caused by missing parts in the input into a machine or manufacturing station?
[P1] What do you mean by missing parts? Parts that disappear for the systems or bad parts?
[R1] If you input parts into a machine for manufacturing, maybe you input them by a tray and the tray is not completely filled, that could cause an error.
[P1] Okay, so the trays we are going to use for this use case are empty trays that will be filled in by the robot and we don't have the cause for missing parts there. All the trays must be 100 percent filled and this is automatic and we don't have missing part there.
[R1] So you don't have this type of exception?
[P1] No, Elena what is your opinion?
[P2] The same, but I am not completely sure if these questions are related in as we are working now or when we will have the robot implemented.
[P1] I think the robot has nothing to do with this process, because the robot will just fill in an empty tray and the whole process will remain the same.
[R1] The questions are focused on exceptions occurring during the SHOP4CF pilots.
[P2] So when the robot will be implemented, yeah, okay. So it is as Oscar mentioned.
[R1] So we can go to the next question. Assuming you have systems or software in place to monitor your manufacturing process, what kind of systems are those?
[P2] We have an MES system in all the lines. You know what an MES system is?
[R2] Gijs may not be very familiar, but I know I can explain to him.
[P2] All the automotive system is in this software system.
[R1] Okay, have you ever encountered any software bugs in these systems?
[P2] What do you mean by software bugs?
[R1] Basically, that the software doesn't do what you intend it to do.
[P1] No we don't have them, we have software engineers here on our plant and our maintenance, and they test this software, they have the software online so they test this frequently, so we don't have bugs. This is something that if it happens, it happens really seldom.
[R1] Okay, sticking with the software, usually this software works with tasks. Have you ever had an exception caused by a task failure in the software?
[P1] Normally, no. We work in the production with a concept that is called {inaudible}, that means that we, before we go into production we make some tests with 30, 40 parts. And we test all the software with every variant. So when we start with a new part number, we test the whole line again, so we don't expect to have during the production any problem with the software. The software is a step activity, it's a PLC program, and we test it always every variant, so we don't have problems normally with that. We also improve this software in order to improve our cycle time, we do always test before we go into production, so if something happens, we will solve it in the testing phase.

[R1] That's it for the software related questions. Next up, if a power outage were to occur, how would this affect your manufacturing process? Do you have some kind of back-up power supply or not?
[P2] Some machines, yes, for 15 minutes or 20 minutes, in order that the computer can lock down okay with the system, but for other machines we don't have any power support. It depends if the machine has a computer or a system that needs it.
[R1] So, basically, your whole manufacturing process will come to a stop if a power outage were to occur?
[P2] yes.
[R1] Okay, so the machines that do have power support ensure that the system shuts down correctly, so when the power comes back on, is it easy for you to restart the system?
[P2] It depends on the machines.
[P1] We have written protocols, because during the Covid situation, we did that really often, because we were able to work in necessary production, so we made protocols to stop the lines and to run it again. So sometimes we face some problems, but with these protocols we minimize this.
[P2] Yeah, exactly. But if we have a power outage, usually the machines are not working with these protocols, so some of these machines need a lot of time to restart, but it depends on the machine.
[R1] Alright, and do you have a system in place that can stop the manufacturing process or part of the manufacturing process in case it detects that a serious error might occur? So it's called a preemptive emergency stop; just in case something bad might happen, it can shut down part of or the whole process. Do you have such a system in place?
[P2] This question I don't understand so well.
[R2] Let's think of an example, so for example a small accident happens, or a small fire or whatever. Is there a physical system that can halt the production or can it be done by some kind of software to stop the production.
[P2] This I don't know
[P1] We have this, I don't know in English, but this red button that can shut everything down in 1 second, because we see that there is a risk to the operators.
[P2] Yeah, but they say a software system. I don't know if we have a software system.
[P1] Software, no.
[R2] So it's a physical way to stop the system.
[P2] Physical, yes, but software, I don't think we have it.
[P1] We have some software systems, it depends on the machine. Because we have some automatic robots, if they detect a force that is not programmed or resistance that is not expected, they shut down automatically, to avoid a crash of the robot. But this is really specific. Normally we don't have something like that.
[R1] Alright, that answers the question rather well. And in your manufacturing process, I assume that a part goes from one step to another. Is it possible for this process to go in the wrong order?
[P1] It is not possible. Because by every production step, we ask the part what has been done before, and if we see this part has not been following the last step before the next step, we scrap the part automatically. We have an electronic system that is called MES, and with this we avoid these kinds of mistakes.
[P2] Yeah, all the parts from the beginning to the end they have traceability, and as Oscar mentioned, if one part loses the traceability, we scrap the part.

[R1] Alright, that's it for the process related exceptions. The machine related exceptions is just one question luckily. How often does one of the machines or a part within a machine or robot break down on average?
[P2] Very often. It's happened all the days in a lot of machines. We have the maintenance department, and they are also in the production, so it is very quick to repair. Well it depends on the breakdown.
[P1] But we work 24 hours in failsafe the whole day, and we use the machines more than 70 percent. So it happens of course, but not that often. We produce more parts than breakdown.
[P2] Yeah for sure.
[P1] We must be over 70 percent always.
[R2] Does this mean that when a machine breaks down you have an alternative to reroute the production, or do you just wait until it's fixed.
[P1] It depends. We have different levels of reactions. First reaction is the operator, we have technical operators also. Next, if they cannot solve it from the first step, second level is the maintenance, third level is the processing linear, fourth level I think is escalation process. Go direct with the team, if not we call the supplier. But normally we don't stop lines that often, we call the supplier for that. This happens really seldom.
[R1] Okay, so what you're saying is that usually it doesn't have that big of an impact if a breakage occurs in one of the machines
[P1] We have always stops of the line, but sometimes it's because a tray has been blocked, or, I don't know, a workpiece carrier cannot be read by the memories and then we have to take out this workpiece carrier. We have a bunch of events. We can see the person takes of the events, and we want to reduce them.
[R2] Well, the next section, the order related, might be a broader scope than let's say the pilot itself, because it refers to the business orders, it might not be very relevant in your case, but we have seen in other pilots it might be interesting. For example in the Volkswagen use case you have order for a car and last minute a customer might want to change the configuration and how does this affect the process. So Gijs will ask these questions, but if it is not very relevant for you, it is fine.
[R1] Yeah, so I'll start. If it is not very relevant, please let me know and we can move on to the next category. Do you have a possibility for supplying a rush order? So an order that has a shorter than usual delivery time.
[P2] It depends on the situation. In the logistic department, they don't offer it. But the clients when they put a rush order, we are trying not to upset. But sometimes, depending on the situation, if they are going to stop the lines, we can accept. But this is not a common situation.
[R1] Alright, and how do these rush orders affect your manufacturing process and the delivery time of your regular orders?
[P2] This I don't know. We shall ask to the logistic department.
[P1] What is the question? Because I had to step out for a minute.
[R1] No problem. We already discussed that there isn't really a possibility, but it does happen sometimes for rush orders to occur.
[P2] A short time order.
[P1] Okay, yes, a short time order. So this is completely fixed in the contract and they cannot do that, and they cannot increase over 50 percent and they cannot do that within 2 weeks. Of course they do that, but we try to solve that situation with them, but the flexibility of production costs a lot of money, so if they pay, we do that. Because we have to do a changeover, because for a customer that costs money. So it's always under negotiation, but they cannot do that regularly.

[R1] Okay, so I just asked when you came back how these rush orders affect your manufacturing process and the delivery time of your regular orders.
[P2] This has an effect, but how? We have to make a changeover in the line in a short time and sometimes it is not possible, depending on the line. But sometimes we need 2 hours to change the machine process so it is not easy, depending on the product and the short time order. If you are ordering for the next day, it is not possible, but if you are ordering for 1 week, yeah, it is more easy for the production department to accept the short time order.
[R1] Alright, and when you get an order, it specifies the details of that order, but how much freedom do you offer to later on change the details of that order. So maybe change the color of a certain part. I don't know if you offer color options, but as an example
[P2] Not in the use case we are speaking, but in other products we have colors, but it is easy for us, depending on if we have one of the parts we need with different colors, and if we have this color we can accept this short time order, but if we don't have that, we need to order to the supplier, and it is not possible for us to do this change.
[R1] Okay, and do you often get orders cancelled? So you get an order, you start working on it and then it gets cancelled, or is that something that doesn't really occur?
[P2] Yeah, we have a lot of cancelled orders.
[P3] Although when production is working, not.
[P2] It depends, this happens also. Sometimes, we produce the parts, and we are finished, and the order is not anymore in the system. So we have a system that we can see when they cancel the part, so when they are not anymore in the system, because they have a program with the client and you can see when they cancel the part. And we try to send the parts, even when the order is not in the system anymore.
[R1] So do these cancelled orders influence your manufacturing process a lot, or not at all?
[P2] No
[R1] Alright, then we can move on to the final category I identified in my study, which is the data flow related exceptions.
[R2] I think some of these questions are quite similar to the machine breakdowns or the software bugs, so maybe Gijs you can see if you have 1 specific one you can ask. Maybe Bosch has to share something else that doesn't fit into one of those categories. I think that is more important to their input.
[R1] Yeah, so I'll just ask 1 main question. You have some sort of data flow between your machines and an information system. How often do errors occur in the flow of data, so that you do not receive data from your machines in your information system, or the other way around?
[P2] How often, I don't know, but sometimes it happens that the machine doesn't send the data flow to the MES system and the machine stops, but how often, I don't know.
[R1] So when this occurs, only the one machine stops, or does it have a bigger influence on your manufacturing process?
[P2] Some machines when this happens the machine stops without any information, and the operator has to take out this part from the line.
[R1] But the rest of your manufacturing process can still continue when this happens?
[P2] Yeah
[P3] The other lines, yes
[R1] So that is my main data flow related question. Are there any other kinds of exceptions that you can think of now, that weren't covered by any of these categories that do occur?
[P3] In production lines many things can happen.
[R1] Yeah, I realize that it is a really broad question, but maybe something that immediately jumps to mind that isn't covered by any of these categories

[P3] It can happen that you have to start a production and one of the raw materials have not been delivered to the factory, or the packaging is not prepared, or the data about the product is not ready, many things. Maybe the operator has gone to the medical center, or whatever.
[P2] Or to the toilet.
[P3] So everything is possible
[R1] Alright, everything is possible, I think that is a nice one to stop my questions with.

Appendix G – List of decision trees of operational exception handling guidelines (Ch. 4 – Exception Handling)

The following pages present the designed decision trees of operational exception handling guidelines per exception type category and MOM KPIs category (Section 4.3.3.2). From the matrix of Table 18, the TPR and QPR decision trees are not presented below as they have been presented in Figure 77 and Figure 78 respectively. For convenience, the legend used in the decision trees is explained again in Figure 168.

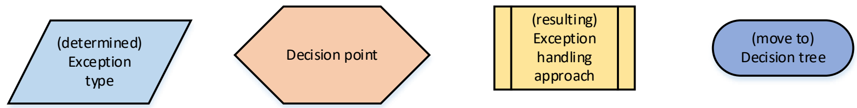


Figure 168: Legend for symbols used in decision trees for selecting suitable exception handling approach.

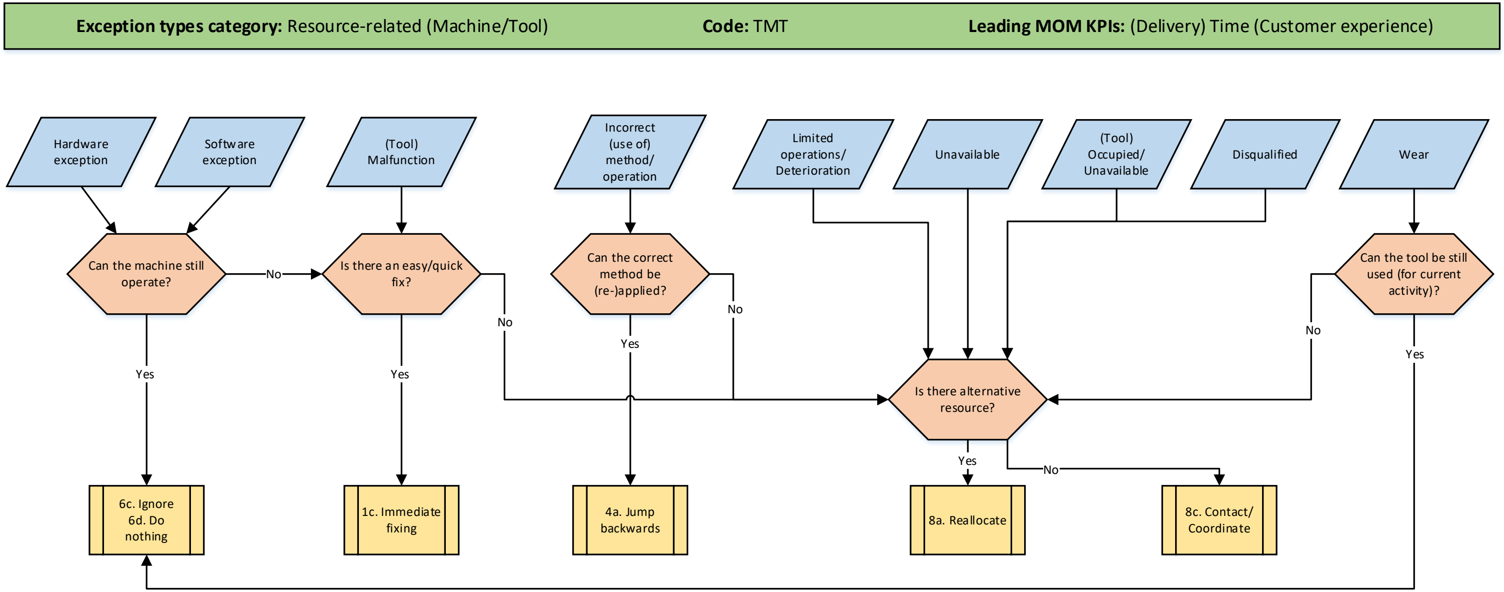


Figure 169: Decision tree for Resource-related (Machine/Tool) exception types categories with (Delivery) Time as the leading MOM KPIs.

Exception types category: Resource-related (Machine/Tool) **Code:** QMT **Leading MOM KPIs:** (Product) Quality

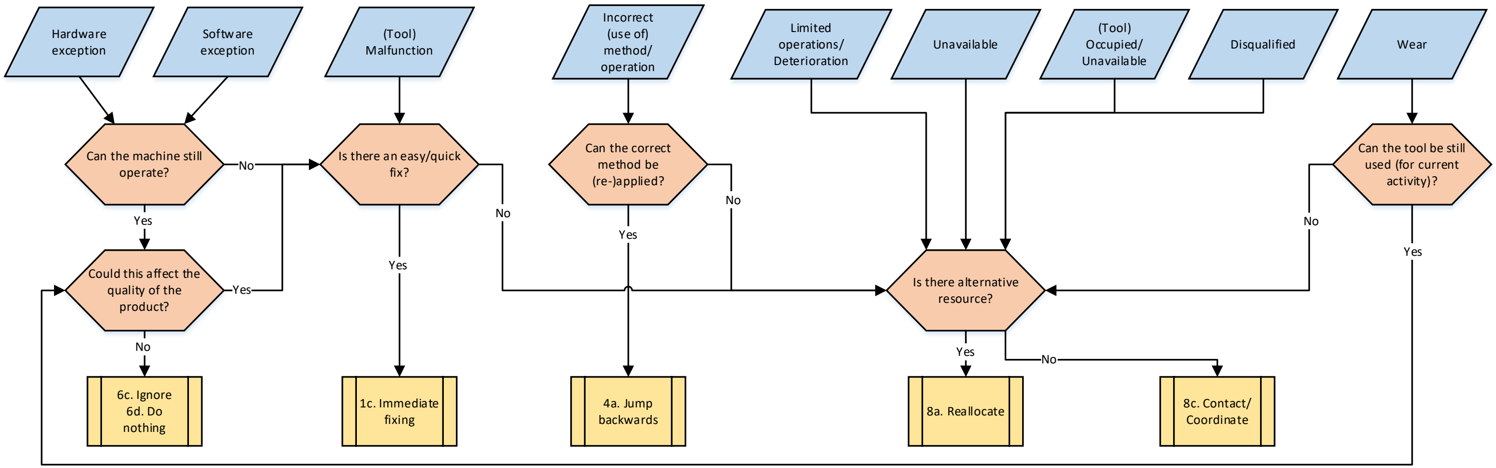


Figure 170: Decision tree for Resource-related (Machine/Tool) exception types categories with (Product) Quality as the leading MOM KPIs.

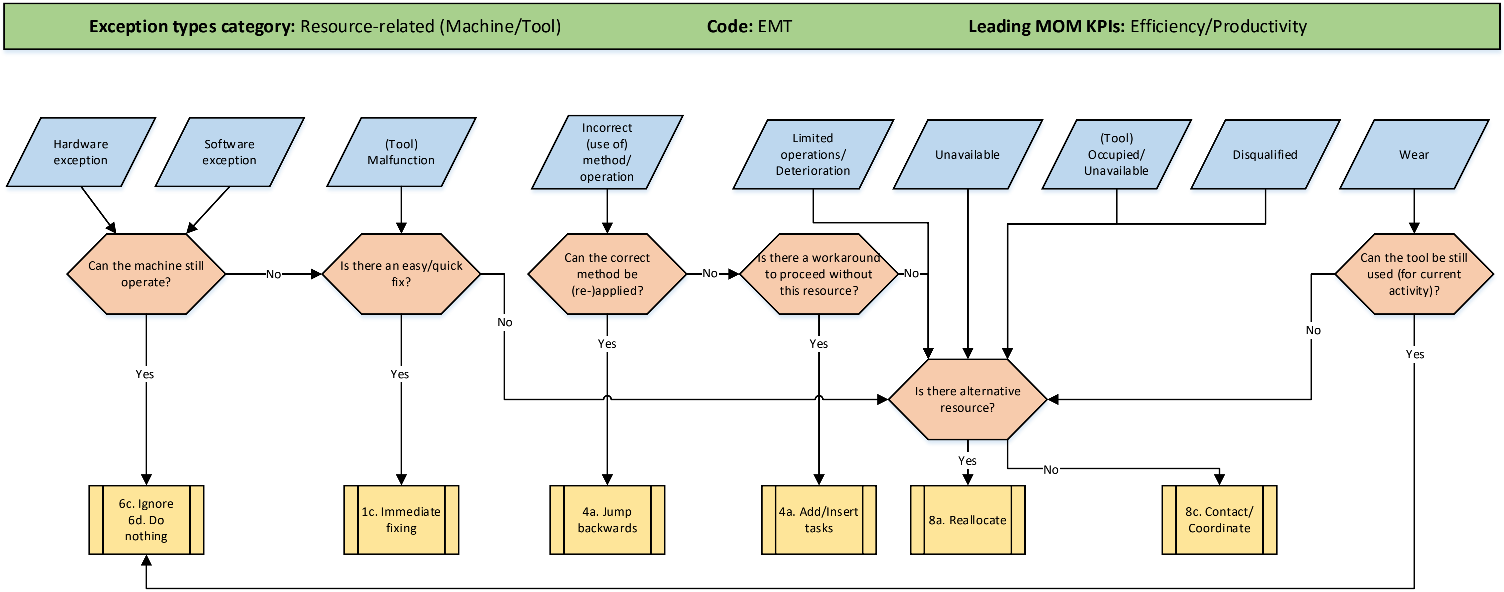


Figure 171: Decision tree for Resource-related (Machine/Tool) exception types categories with Efficiency/Productivity as the leading MOM KPIs.

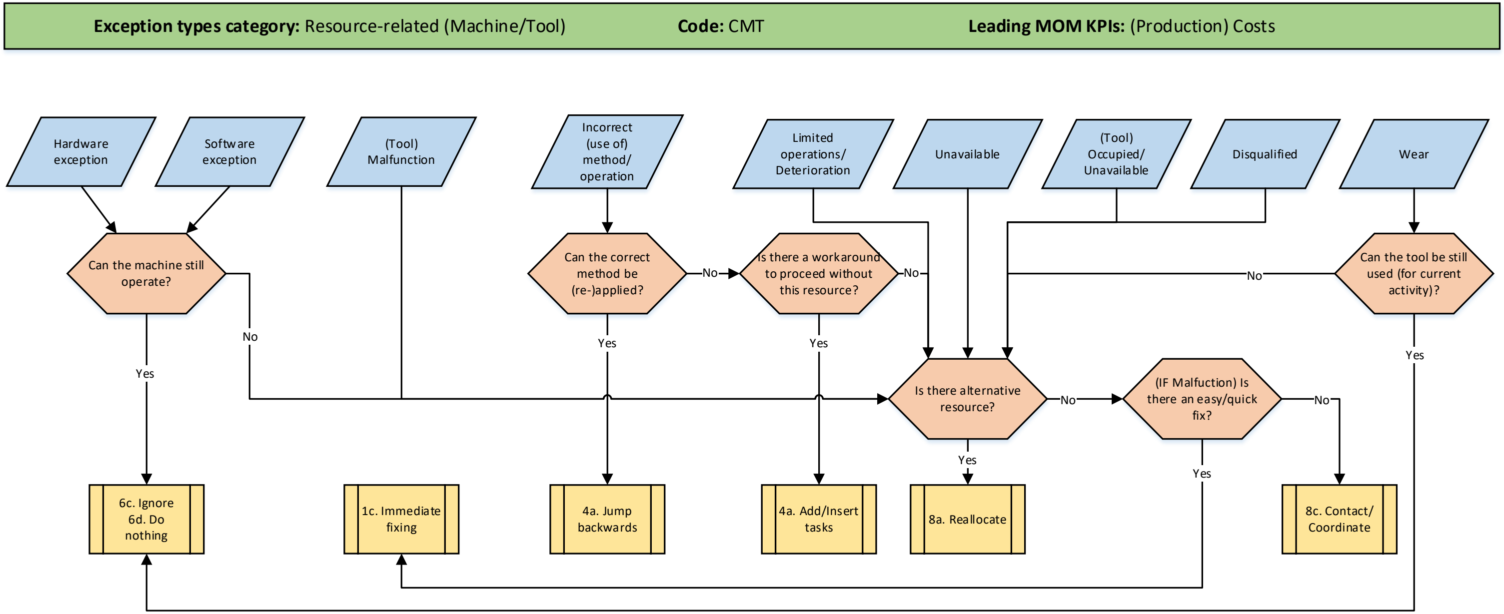


Figure 172: Decision tree for Resource-related (Machine/Tool) exception types categories with (Production) Costs as the leading MOM KPIs.

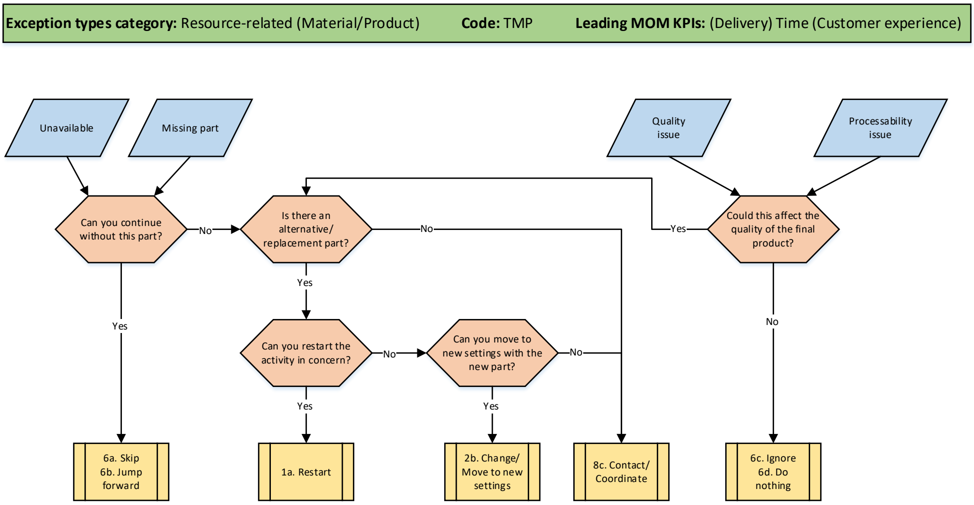


Figure 173: Decision tree for Resource-related (Material/Product) exception types categories with (Delivery) Time as the leading MOM KPIs.

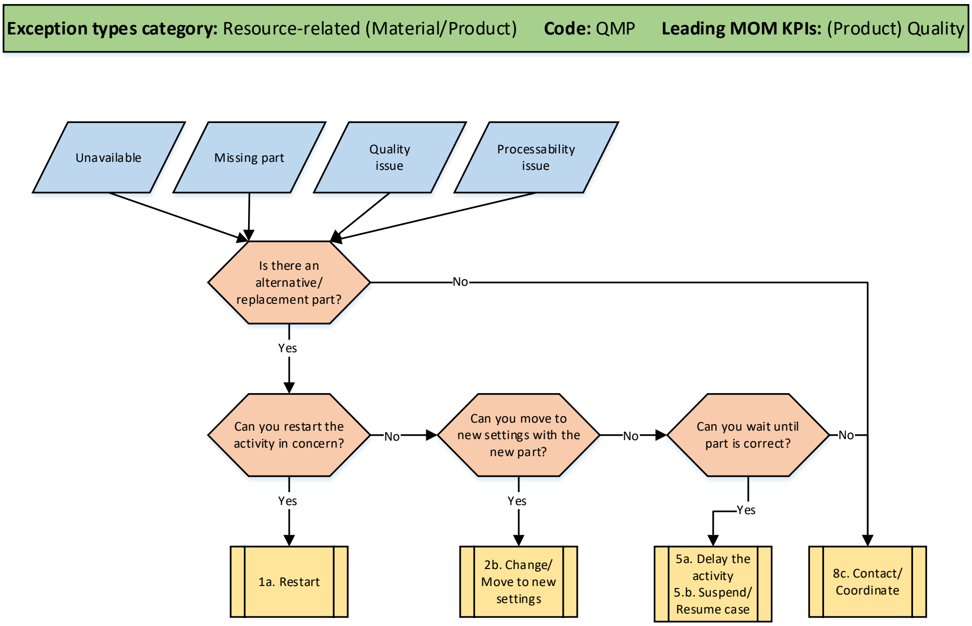


Figure 174: Decision tree for Resource-related (Material/Product) exception types categories with (Product) Time as the leading MOM KPIs.

Exception types category: Resource-related (Material/Product) **Code:** EMP **Leading MOM KPIs:** Efficiency/Productivity

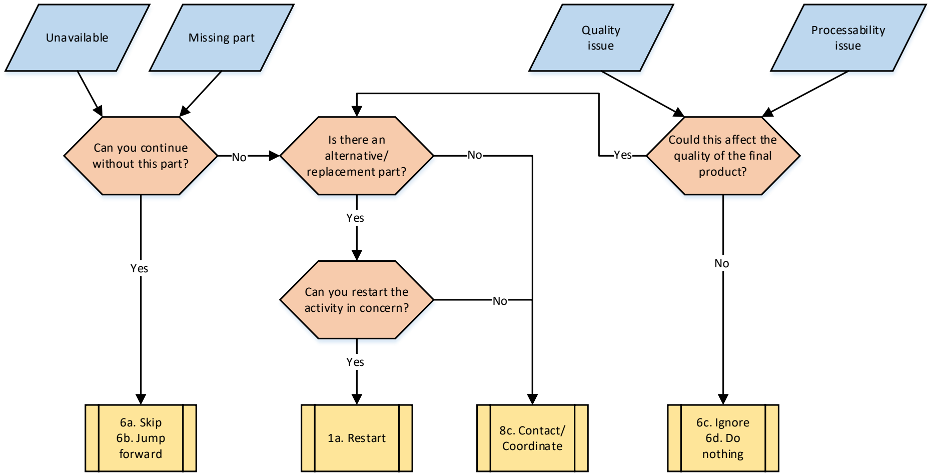


Figure 175: Decision tree for Resource-related (Material/Product) exception types categories with Efficiency/Production as the leading MOM KPIs.

Exception types category: Resource-related (Material/Product)
Code: CMP **Leading MOM KPIs:** (Production) Costs

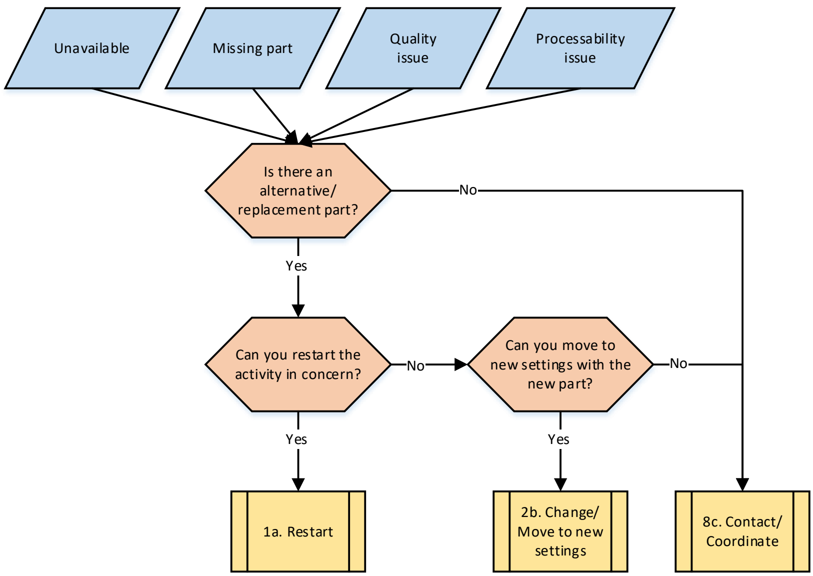


Figure 176: Decision tree for Resource-related (Material/Product) exception types categories with Efficiency/Production as the leading MOM KPIs.

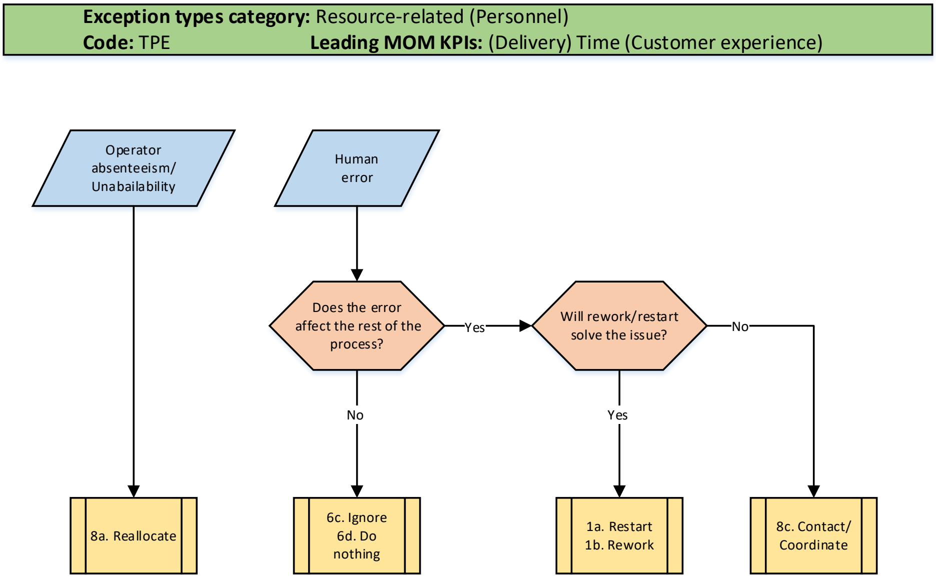


Figure 177: Decision tree for Resource-related (Personnel) exception types categories with (Delivery) Time as the leading MOM KPIs.

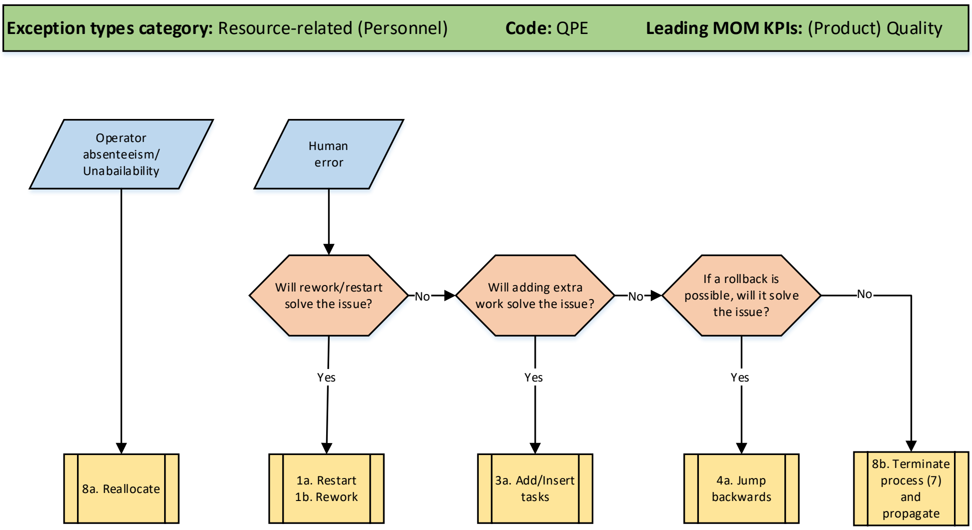


Figure 178: Decision tree for Resource-related (Personnel) exception types categories with (Product) Quality as the leading MOM KPIs.

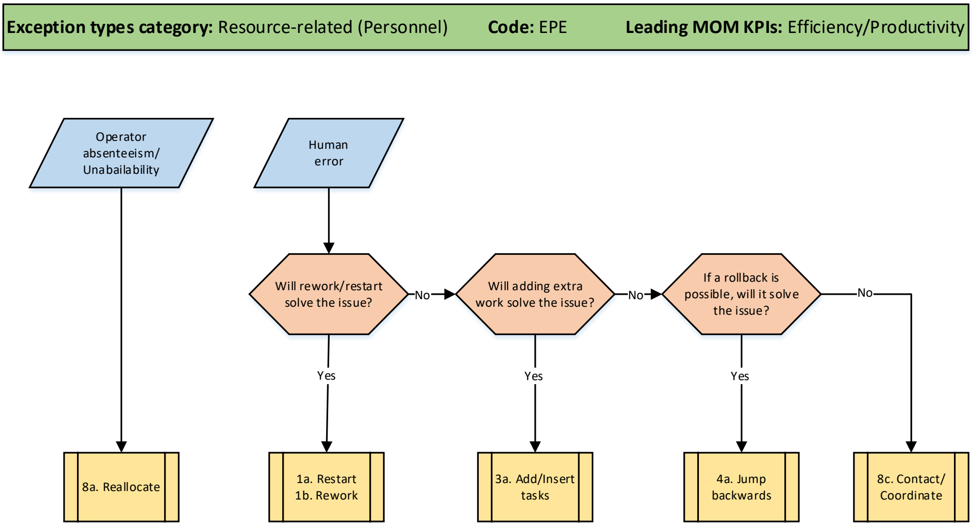


Figure 179: Decision tree for Resource-related (Personnel) exception types categories with Efficiency/Productivity as the leading MOM KPIs.

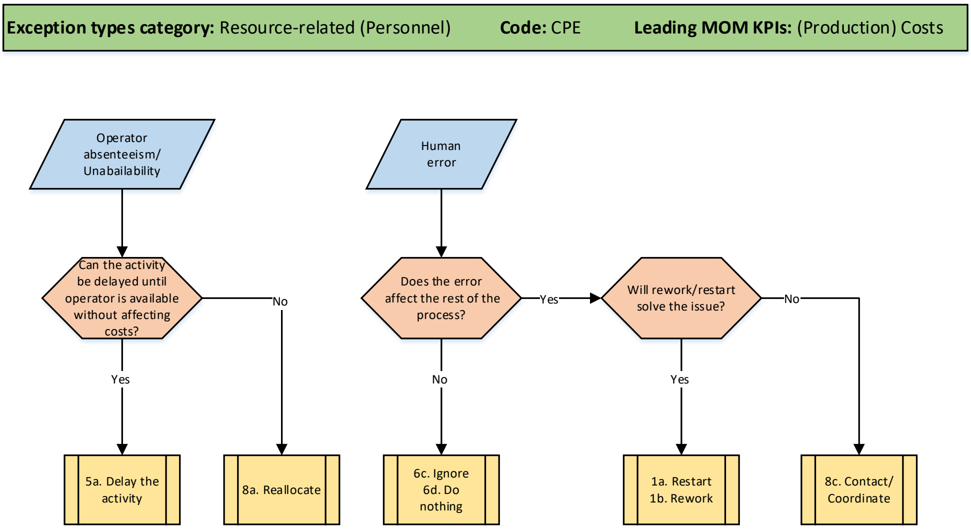


Figure 180: Decision tree for Resource-related (Personnel) exception types categories with (Production) Costs as the leading MOM KPIs.

Exception types category: Resource-related (Infrastructure) **Code:** INF **MOM KPIs:** Time/Quality/Efficiency/Costs

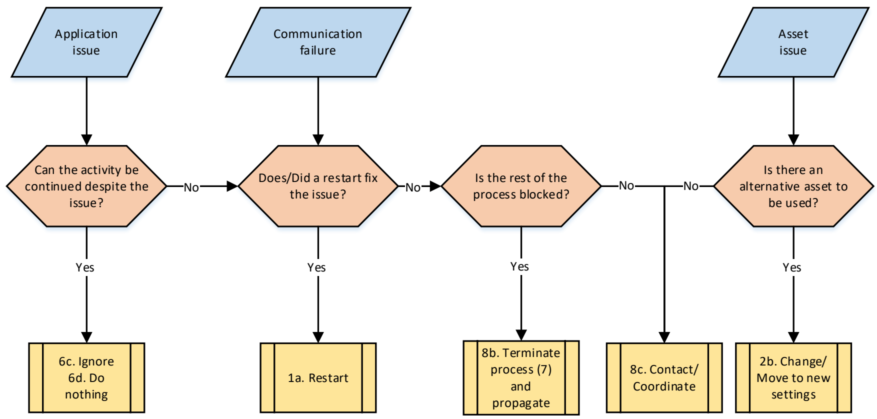


Figure 181: Decision tree for Resource-related (Infrastructure) exception types categories for all MOM KPIs (Time/Quality/Efficiency/Costs).

Exception types category: Order-related **Code:** TOR **MOM KPIs:** (Delivery) Time (Customer experience)

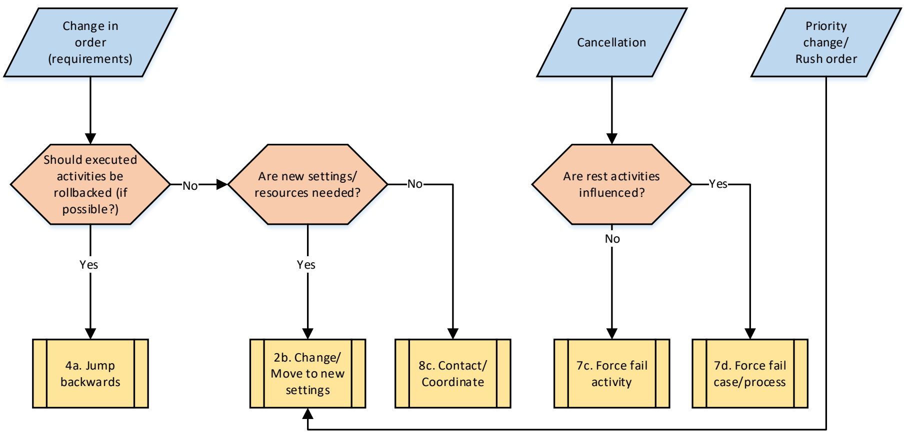


Figure 182: Decision tree for Order-related exception types categories with (Delivery) Time as the leading MOM KPIs.

Exception types category: Order-related **Code:** EOR **MOM KPIs:** Efficiency/Productivity

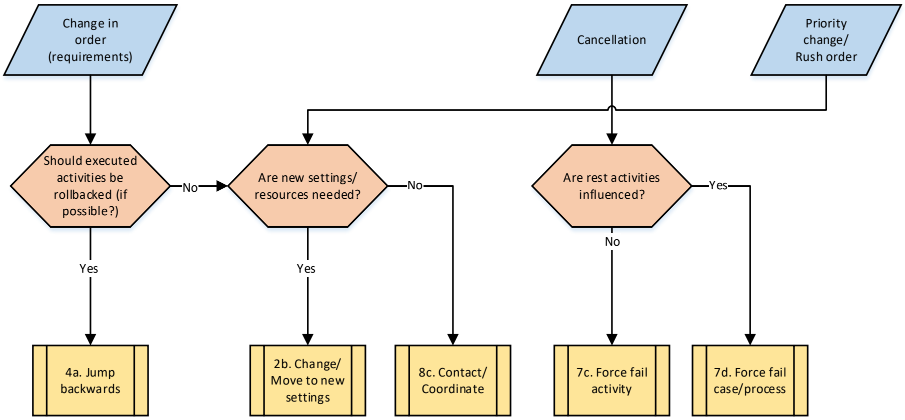


Figure 183: Decision tree for Order-related exception types categories with Efficiency/Productivity as the leading MOM KPIs.

Exception types category: Order-related **Code:** COR **MOM KPIs:** (Production) Costs

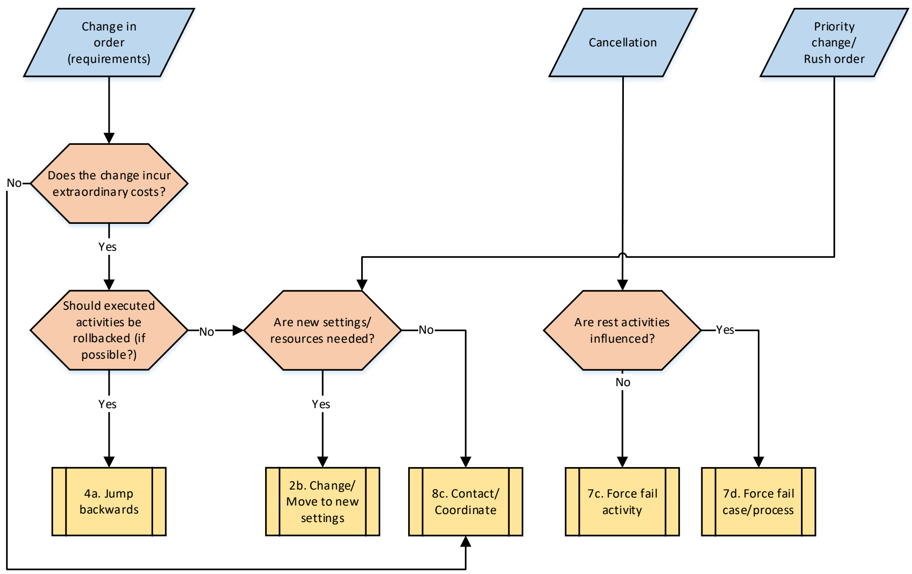


Figure 184: Decision tree for Order-related exception types categories with (Production) Costs as the leading MOM KPIs.

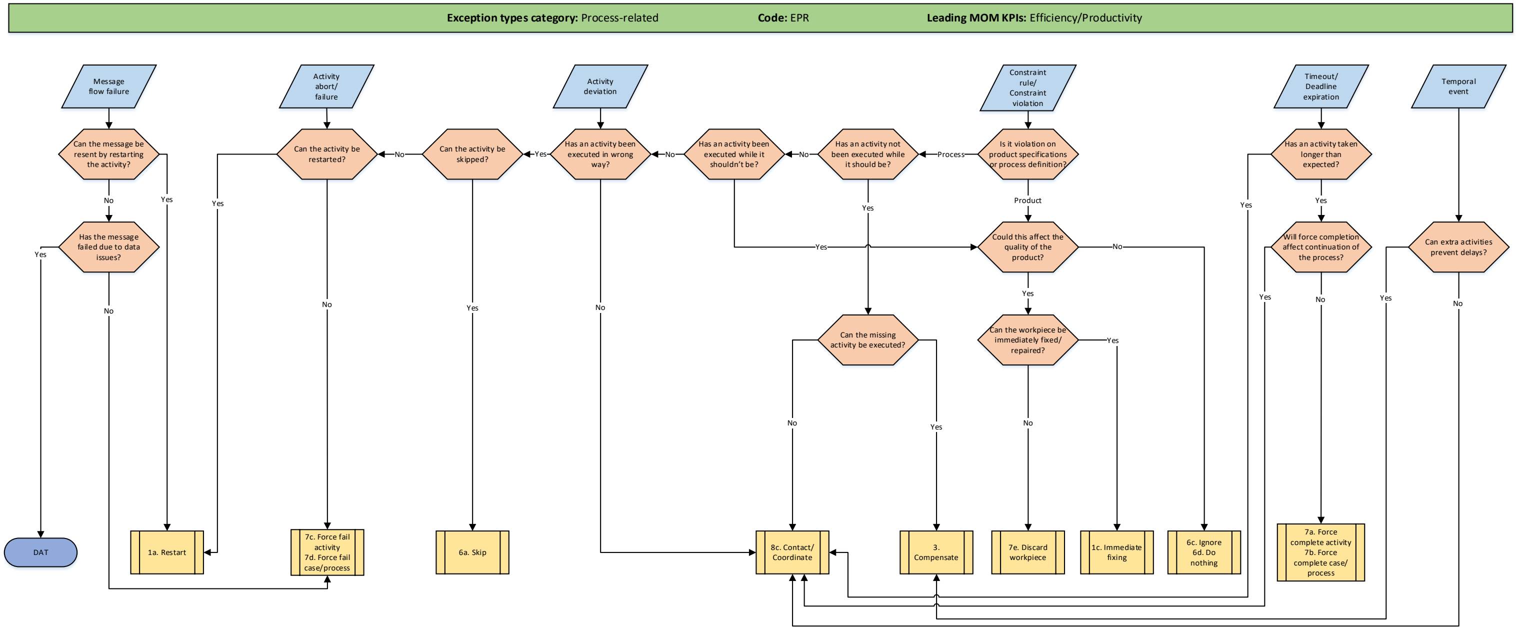


Figure 185: Decision tree for Process-related exception types categories with Efficiency/Productivity as the leading MOM KPIs.

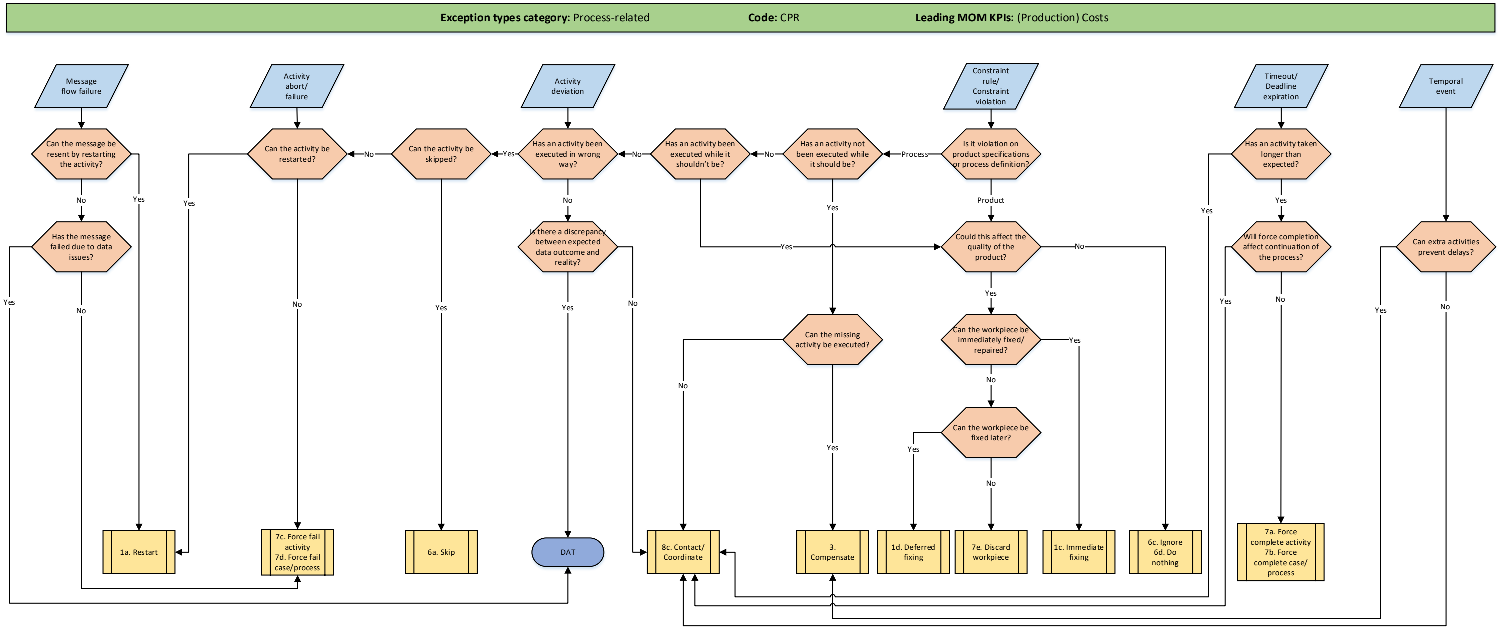


Figure 186: Decision tree for Process-related exception types categories with (Production) Costs as the leading MOM KPIs.

Exception types category: Event-related **Code:** TEV **MOM KPIs:** (Delivery) Time (Customer experience)

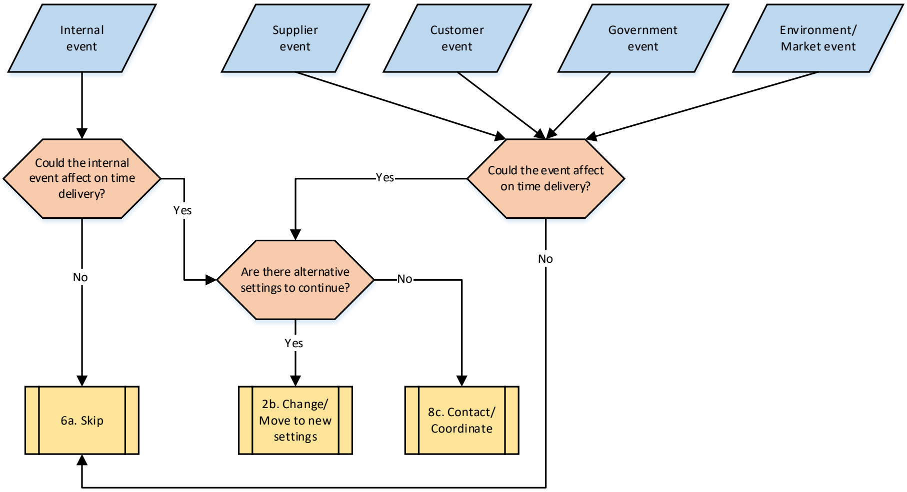


Figure 187: Decision tree for Event-related exception types categories with (Delivery) Time as the leading MOM KPIs.

Exception types category: Event-related **Code:** QEV **MOM KPIs:** (Product) Quality

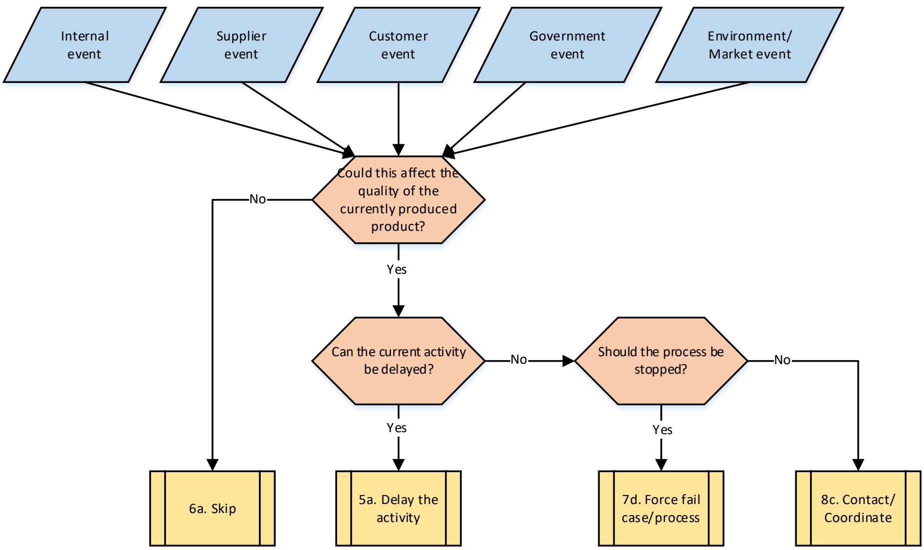


Figure 188: Decision tree for Event-related exception types categories with (Product) Quality as the leading MOM KPIs.

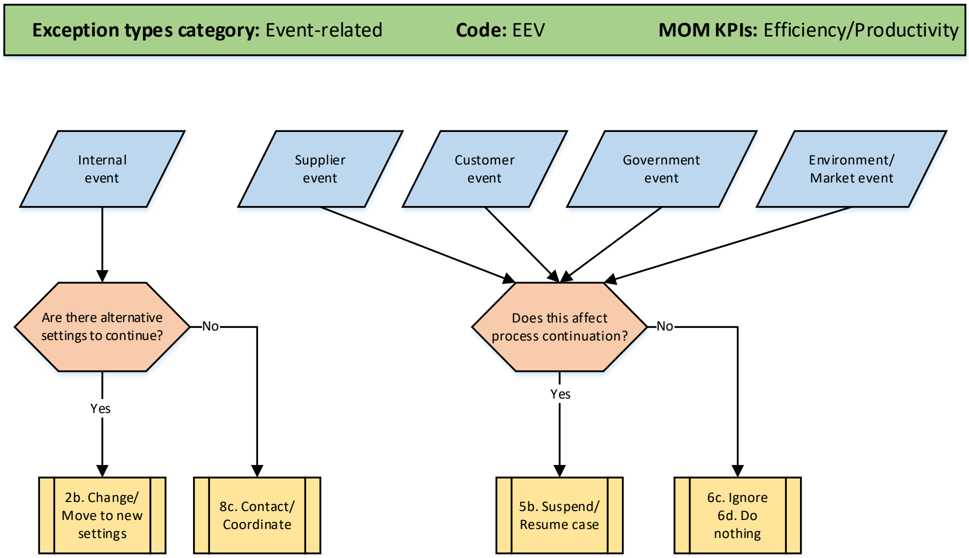


Figure 189: Decision tree for Event-related exception types categories with Efficiency/Productivity as the leading MOM KPIs.

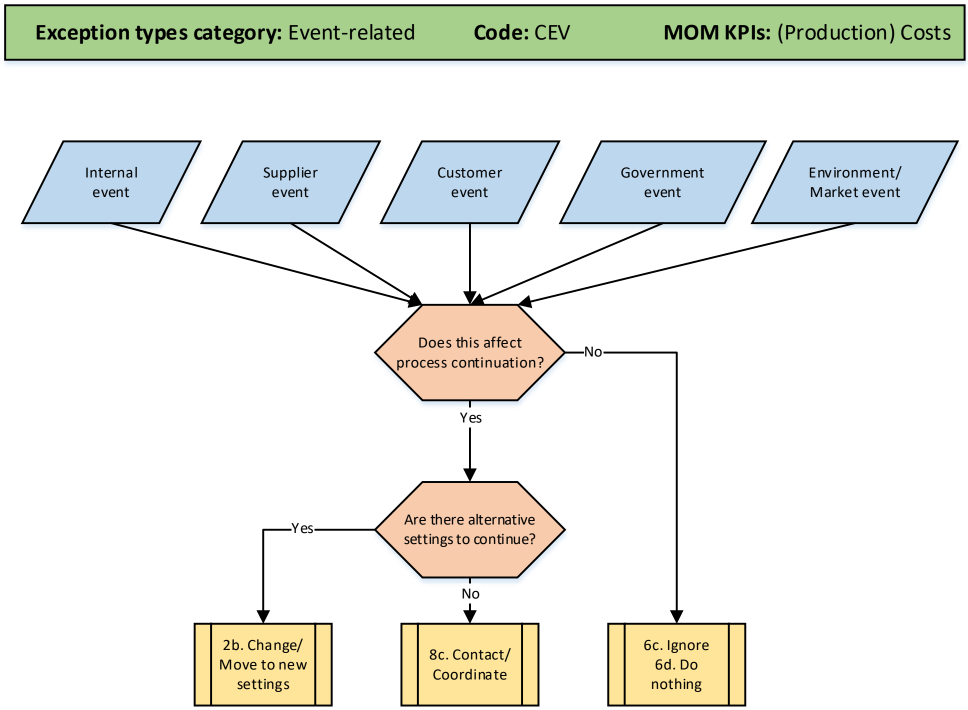


Figure 190: Decision tree for Event-related exception types categories with (Production) Costs as the leading MOM KPIs.

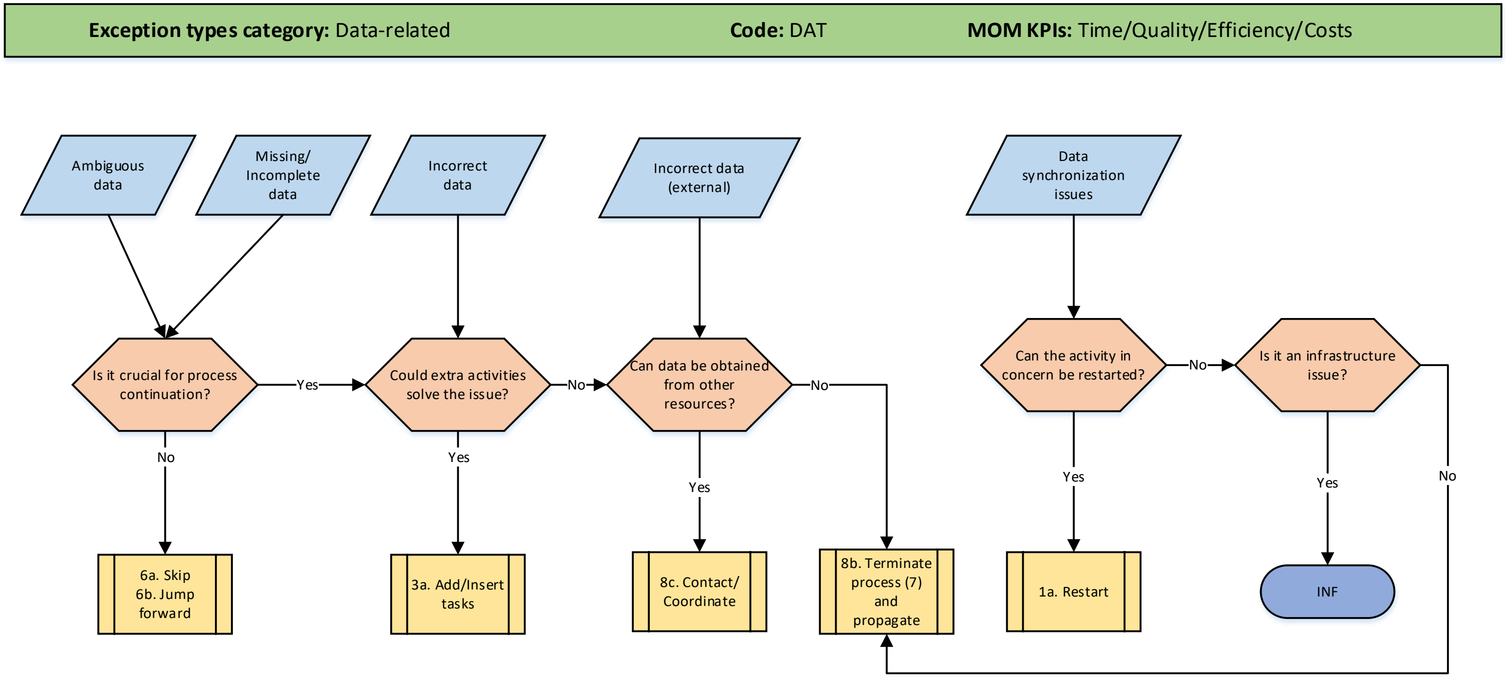


Figure 191: Decision tree for Data-related exception types categories for all MOM KPIs (Time/Quality/Efficiency/Costs).

Appendix H – FIWARE Smart Industry reference architecture

Figure 192 presents the FIWARE Smart Industry reference architecture. It is the selected middleware technology in SHOP4CF project.

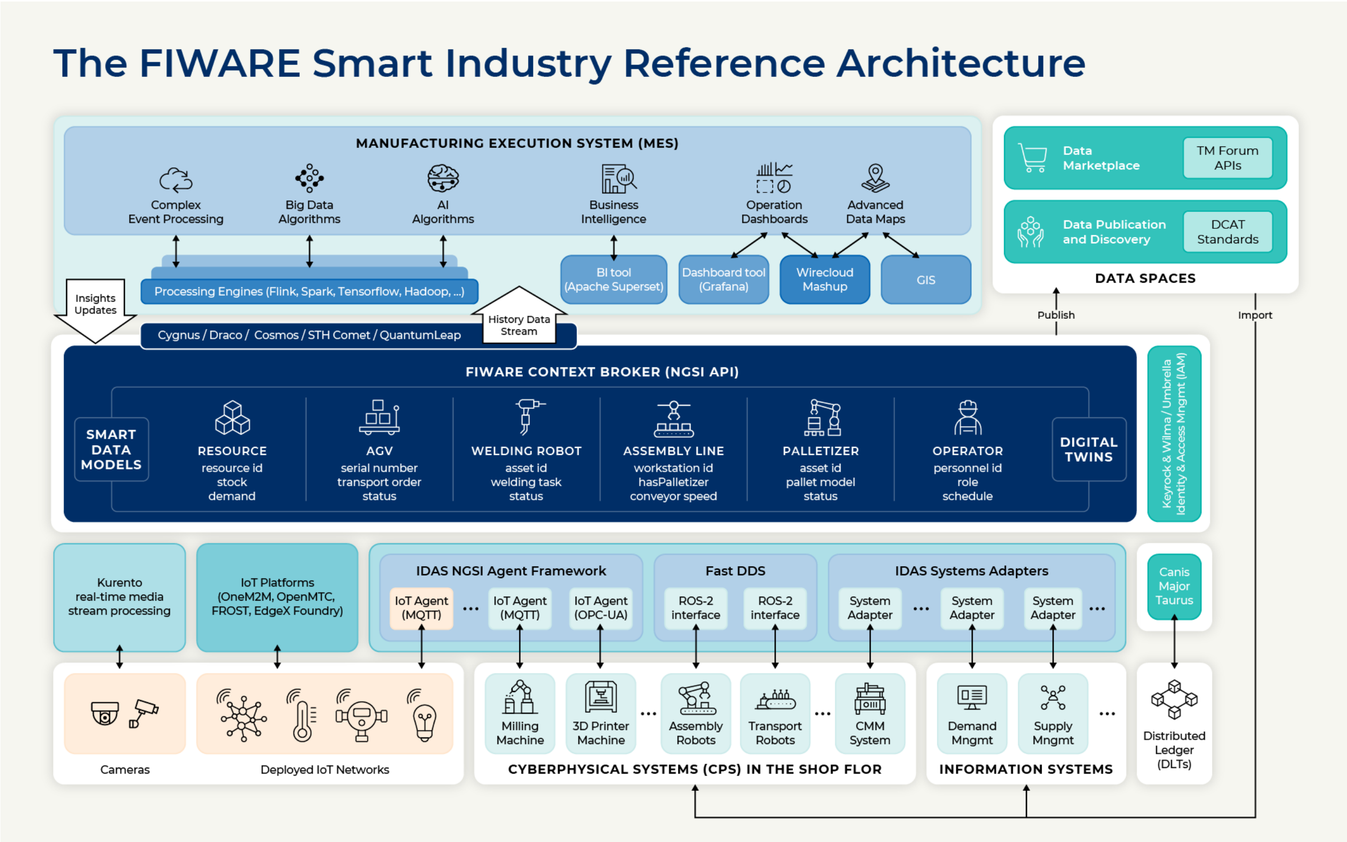


Figure 192: FIWARE Smart Industry reference architecture[92].

[92] https://www.fiware.org/community/smart-industry/

Appendix I – Recipe system data model

This appendix presents the data model of the operationalized recipe system, discussed in Section 6.2.1.2. The complete data model class diagram is shown in Figure 193.

The Recipe class represents the recipe notion. It manages multiple pools and may contain StartDefinition (necessary when a recipe fulfilment triggers the execution of a start event/new process). The Pool class uses an AvailabilityMask implementation to determine which instances are available. Process instances are ranked through the ReleasePolicy object. A BufferedInstance class contains information on the process instances that are inserted in a pool. Classes illustrated in gray color are not part of the implementation of the recipe system but are part of the process engine the system integrates to (and thus are added for completeness). The rest classes are auxiliary classes used for message correlation.

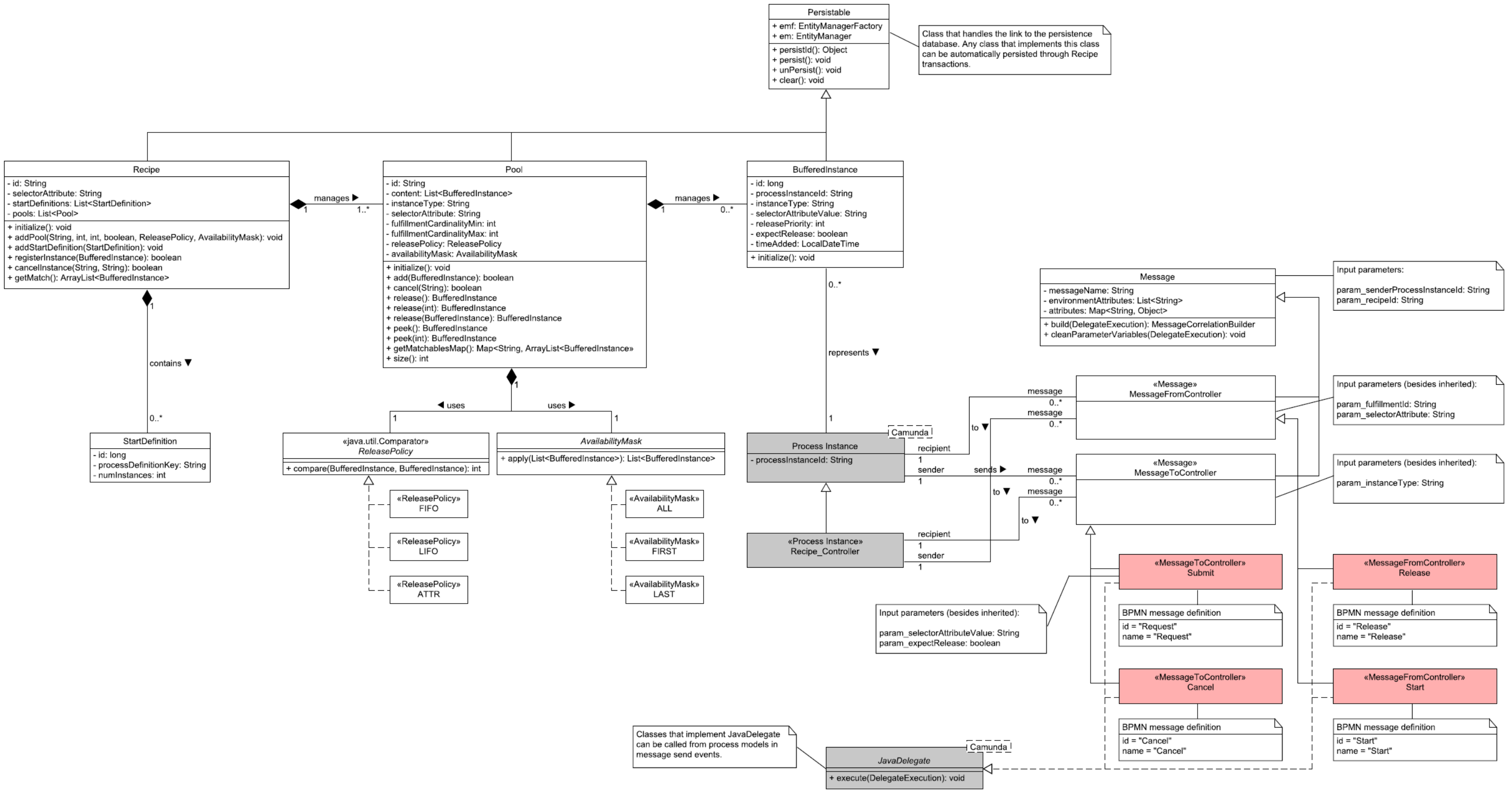


Figure 193: Recipe system data model (Spijkers, 2019).

Appendix J – References information

This appendix provides reference links to implemented solutions and to demonstration media.

Table 39 provides information on accessing the source code of the implemented solutions.

Table 39: Reference links to source code of the implemented solutions.

Link	Description
https://gitlab.tue.nl/is-mpms/horse-mpms	MPMS as implemented for the HORSE project. Includes integration to a web-socket message bus middleware.
https://gitlab.tue.nl/is-mpms/eit-oedipus-ppms	Printing Process Management System (PPMS) as implemented for the EIT OEDIPUS project.
https://gitlab.tue.nl/is-mpms/shop4cf-mpms	MPMS as implemented for SHOP4CF project. Includes integration to Orion-LD context broker (FIWARE).
https://github.com/ramp-eu/Manufacturing_Process_Management_System	Basic version of MPMS as implemented for SHOP4CF project. Project's official repository to be downloaded by interested factories. Includes integration to Orion-LD context broker (FIWARE).

Table 40 provides evidence information from the demonstrated solutions at projects pilots (numbered as per Table 32, Table 33, and Table 34).

Table 40: Reference links to demonstrated media.

Link	Description	Pilot case
https://www.youtube.com/watch?v=gdgRm6DkAyc&feature=youtu.be	*HORSE - MPMS demo (Manufacturing Process Management System)* Introduction to MPMS.	-
https://www.youtube.com/watch?v=hD1vqzykLkU http://www.horse-project.eu/sites/default/files/videos/HORSEglobal.mp4	*Manufacturing Process Management with Robot Task Synchronization* Orchestration of automated agents (robotic arm and camera) for product visual inspection.	HP1
https://www.youtube.com/watch?v=bqTDEZvOdVI&feature=youtu.be	*Pilot case 1 demonstration* Tool assembly process at TRI pilot, assisted by AR and mobile robot for parts fetching.	HP2
https://is.ieis.tue.nl/edoc20/videos/Kostas%20Traganos.mp4	*End-to-End Production Process Orchestration for Smart Printing Factories: An Application in Industry* EDOC 2020 paper presentation on MPM developments in production printing.	EO1
https://surfdrive.surf.nl/files/index.php/s/HhVhUsg4wxfarT1	*Video on BOS-S pilot* Trays feeding process with mobile robot, orchestrated by advanced MPMS.	SP1

Appendix K – Deployed MPM solutions per real-world pilot case

Table 41 maps per pilot (numbered as per Table 32, Table 33, and Table 34) the deployed solutions developed in the frames of the current research. The deployment was performed based on the identified challenges (i.e., whether the proposed solutions fit in the scope of a challenge) and feasibility (e.g., time, budget, priorities).

Table 41: Deployed MPM solution per pilot matrix.

Developed solution		**Real-world pilot cases (HORSE – EIT OEDIPUS – SHOP4CF projects)**													
Aspect	**Detailed functionality**	**HP1**	**HP2**	**HP3**	**HE1**	**HE2**	**HE3**	**HE4**	**HE5**	**HE6**	**HE7**	**EO1**	**SP1**	**SP2**	**SP3**
Flexible modeling	Task delivery modeling patterns	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	Human-robot modeling patterns		X											X	
	Synchronization mechanism											X			
Exception handling	Exception handling modeling constructs	X	X	X								X	X	X	
MPMS	Process Modeler	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	Execution Engine	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	CPS integration	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Appendix L – Advanced MPMS evaluation form

This appendix presents the evaluation form used in the semi-structured interviews for assessing advanced MPMS developments (as discussed in Section 7.2).

1. Introduction

You are invited to participate in this survey with the purpose to evaluate the MPMS component. The survey is structured as follows:

1. Background information on your profile is collected, for assessing the suitability of the target participants.
2. Background information on MPMS is provided, to summarize its application purpose and functionality.
3. Evaluation of statements on perceived ease of use of MPMS.
4. Evaluation of statements on perceived usefulness of MPMS.
5. Evaluation of statements on intention to use MPMS.
6. Open questionnaire.

Note that this survey is not a test of knowledge or skills. There are no right or wrong answers. Please fill in the questions as realistically as possible.

The questionnaires consist of closed and open questions. Closed questions are answered by selecting one of the 5 available options. An example is given below.

		5-point likelihood scale				
Statement		1 - Strongly agree		3 - Neutral		5 - Strongly disagree
1	I like coffee better than tea.	☒	☐	☐	☐	☐
2	I do like sugar in my coffee.	☐	☐	☐	☒	☐

1.1 Privacy disclaimer

The data collected through this survey is processed anonymously and is under no circumstances used to trace answers back to individuals. The results are exclusively published in aggregated form, with the sole purpose of evaluating this research's proposed solution artifacts.

2. Participant's profile

- My current job title is: ……………………………………

- My current job is best classified as (pick one):

☐ (Information) Systems Architect
☐ Developer
☐ Product/System Designer
☐ Operations Manager
☐ Integrator
☐ Robotics expert
☐ Other (please specify): …………………………………

Please indicate to which degree you agree with the following statements with respect to your background knowledge in Business Process Management (BPM):

	Statement	5-point likelihood scale				
		1 - Strongly agree		3 - Neutral		5 - Strongly disagree
1	I am familiar with BPM techniques and tools.	☐	☐	☐	☐	☐
2	I am interested in applying BPM techniques and tools.	☐	☐	☐	☐	☐
3	I consider myself expert in BPM.	☐	☐	☐	☐	☐
4	I am interested in process/workflow modeling.	☐	☐	☐	☐	☐
5	I am familiar with BPMN modeling.	☐	☐	☐	☐	☐
6	I consider myself expert in BPMN modeling.	☐	☐	☐	☐	☐

3. MPMS background info

MPMS is based on Business Process Management (BPM) theories and tools. It is used for process modeling and automated orchestration of activities. It provides horizontal integration by providing cross-functional control (i.e., across production lines and work cells). It provides vertical control by triggering actions by humans and automated actors.

Please refer to the attached presentation on explanation on the modules of MPMS.

4. Perceived ease of use

Please indicate to which degree you agree with the following statements with respect to the perceived ease of use of the system.

	Statement	1 - Strongly agree		3 - Neutral		5 - Strongly disagree
	Perceived Ease of Use					
1.	Modeling processes with MPMS modeler is easy for me.	☐	☐	☐	☐	☐
2.	I find it difficult to provide definitions of involved entities (i.e., resources, tasks, location points) through MPMS.	☐	☐	☐	☐	☐
3.	I find that implementing any business logic (through coding an application project) is difficult for me.	☐	☐	☐	☐	☐
4.	The configuration and customization of MPMS is easy for me.	☐	☐	☐	☐	☐
5.	My interaction with the MPMS applications (Tasklist, Cockpit, Admin) is clear and understandable.	☐	☐	☐	☐	☐
6.	I find it takes a lot of effort to become skillful at using MPMS modules.	☐	☐	☐	☐	☐
7.	I find MPMS rigid and inflexible to use.	☐	☐	☐	☐	☐
8.	Overall, I find MPMS easy to use.	☐	☐	☐	☐	☐

(5-point likelihood scale)

5. Perceived usefulness

Please indicate to which degree you agree with the following statements with respect to the perceived usefulness of the system.

	Statement	1 - Strongly agree		3 - Neutral		5 - Strongly disagree
	Perceived Usefulness					
1.	Using MPMS (Modeler) allows me to clearly represent (production) processes.	☐	☐	☐	☐	☐
2.	With MPMS (Modeler) it is possible to model complex production scenarios.	☐	☐	☐	☐	☐
3.	MPMS allows for process integration and automation that would be not possible (or difficult to achieve) without this system.	☐	☐	☐	☐	☐
4.	MPMS would enable higher productivity through process management.	☐	☐	☐	☐	☐
5.	With MPMS I get a clear overview of (production) processes.	☐	☐	☐	☐	☐
6.	Overall, I find MPMS useful for tackling process complexity.	☐	☐	☐	☐	☐

(5-point likelihood scale)

6. Intention to use

Please indicate to which degree you agree with the following statements with respect to the intention to use the system.

	Statement	1 - Strongly agree		3 - Neutral		5 - Strongly disagree
		5-point likelihood scale				
	Intention to use					
1.	I would consider using the MPMS solutions for tackling process complexity in my organization.	☐	☐	☐	☐	☐
2.	I intend to use MPMS for tackling process complexity in my organization.	☐	☐	☐	☐	☐

7. Open questionnaire

1. From what you have learned from presentations/documentation of MPMS and the demonstrated application of the component, what do you consider as the main positive and negative points?
   - Positive:
     ………………………………………………………………………………
     ………………………………………………………………………………
     ………………………………………………………………………………
     ………………………………………………………………………………
     ……………………………………………………
   - Negative:
     ………………………………………………………………………………
     ………………………………………………………………………………
     ………………………………………………………………………………
     ………………………………………………………………………………
     ……………………………………………………

2. Do you have any similar system(s) (e.g., MES) that cover(s) (part) of MPMS functionality?
   - If yes, please indicate what type of system(s) cover(s):
     - Process modeling:
       ………………………………………………………………………
     - Automated process execution:
       ………………………………………………………………………
       ……………

- Task delivery to humans and automated actors: …………………………………………………………………………………
- Exception handling: ……………………………………………………………..............
- Process monitoring: …………………………………………………………………
- o No:……..

3. What functionality would you suggest that MPMS should cover, that currently the system misses?
……

4. Which MPMS module do you consider difficult to use?
……

5. Any other comments/suggestions on the MPMS approach?
……

Appendix M – Evaluation results on advanced MPMS utility

This appendix presents the evaluation results from the semi-structured interviews with practitioners on usability aspects of advanced MPMS (as discussed in Section 7.2).

Practitioner 1

[redacted] – [redacted]

May 20th
13:00

Manufacturing Process Management System (MPMS)

Evaluation form

1 Introduction

You are invited to participate in this survey with the purpose to evaluate the MPMS component. The survey is structured as follows:

1. Background information on your profile is collected, for assessing the suitability of the target participants.
2. Background information on MPMS is provided, to summarize its application purpose and functionality.
3. Evaluation of statements on perceived ease of use of MPMS.
4. Evaluation of statements on perceived usefulness of MPMS.
5. Evaluation of statements on intention to use MPMS.
6. Open questionnaire.

Note that this survey is not a test of knowledge or skills. There are no right or wrong answers. Please fill in the questions as realistically as possible.

The questionnaires consist of closed and open questions. Closed questions are answered by selecting one of the 5 available options. An example is given below.

		5-point likelihood scale				
Statement		Strongly agree		Neutral		Strongly disagree
1	I like coffee better than tea.	☒	☐	☐	☐	☐
2	I do like sugar in my coffee.	☐	☐	☐	☒	☐

1.1 Privacy disclaimer

The data collected through this survey is processed anonymously and is under no circumstances used to trace answers back to individuals. The results are exclusively published in aggregated form, with the sole purpose of evaluating this research's proposed solution artifacts.

2 Participant's profile

- My current job title is: ...project coordinator...
- My current job is best classified as (pick one):

☐ (Information) Systems Architect
☐ Developer
☐ Product/System Designer
☐ Operations Manager

☒ Integrator
☐ Robotics expert
☐ Other (please specify): ....

Please indicate to which degree you agree with the following statements with respect to your background knowledge in Business Process Management (BPM):

1. examples?
BPMN is common in different domains like banking. K. is trying to use it in manufacturing where other notations (UML) are common.

		5-point likelihood scale				
	Statement	Strongly agree		Neutral		Strongly disagree
1	I am familiar with BPM techniques and tools.	☐	☐	☐	X	☐
2	I am interested in applying BPM techniques and tools.	X	☐	☐	☐	☐
3	I consider myself expert in BPM.	☐	☐	☐	☐	X
4	I am interested in process/workflow modeling.	X	☐	☐	☐	☐
5	I am familiar with BPMN modeling.	☐	☐	☐	X	☐
6	I consider myself expert in BPMN modeling.	☐	☐	☐	☐	X

3 MPMS background info

MPMS is based on Business Process Management (BPM) theories and tools. It is used for process modeling and automated orchestration of activities. It provides horizontal integration by providing cross-functional control (i.e., across production lines and work cells). It provides vertical control by triggering actions by humans and automated actors.

Please refer to the attached presentation on explanation on the modules of MPMS.

4 Perceived ease of use

Please indicate to which degree you agree with the following statements with respect to the perceived ease of use of the system.

		5-point likelihood scale				
	Statement	Strongly agree		Neutral		Strongly disagree
	Perceived Ease of Use					
1.	Modeling processes with MPMS modeler is easy for me.	X	☐	☐	☐	☐
2.	I find it difficult to provide definitions of involved entities (i.e., resources, tasks, location points) through MPMS.	☐	☐	☐	☐	X
3.	I find that implementing any business logic (through coding an application project) is difficult for me.	☐	☐	☐	☐	X

with a tool

4.	The configuration and customization of MPMS is easy for me.	☒	☐	☐	☐	☐	how to run process configuration with dashboard,
5.	My interaction with the MPMS applications (Tasklist, Cockpit, Admin) is clear and understandable.	☒	☐	☐	☐	☐	UI is ok
6.	I find it takes a lot of effort to become skillful at using MPMS modules.	☐	☒	☐	☐	☐	as far as I saw MPMS it was ok
7.	I find MPMS rigid and inflexible to use.	☐	☐	☐	☐	☒	because modifications were done by K.
8.	Overall, I find MPMS easy to use.	☒	☐	☐	☐	☐	

5 Perceived usefulness

Please indicate to which degree you agree with the following statements with respect to the perceived usefulness of the system.

		5-point likelihood scale					
	Statement	Strongly agree		Neutral		Strongly disagree	
	Perceived Usefulness						
1.	Using MPMS (Modeler) allows me to clearly represent (production) processes.	☒	☐	☐	☐	☐	
2.	With MPMS (Modeler) it is possible to model complex production scenarios.	☐	☒	☐	☐	☐	
3.	MPMS allows for process integration and automation that would be not possible (or difficult to achieve) without this system.	☒	☐	☐	☐	☐	in 7
4.	MPMS would enable higher productivity through process management.	☒	☐	☐	☐	☐	
5.	With MPMS I get a clear overview of (production) processes.	☒	☐	☐	☐	☐	
6.	Overall, I find MPMS useful for tackling process complexity.	☒	☐	☐	☐	☐	

6 Intention to use

Please indicate to which degree you agree with the following statements with respect to the intention to use the system.

		5-point likelihood scale					
	Statement	Strongly agree		Neutral		Strongly disagree	
	Intention to use						
1.	I would consider using the MPMS solutions for tacking process complexity in my organization.	☐	☐	☒	☐	☐	[redacted]'s evaluating if MPMS is useful and compliant with rules
2.	I intend to use MPMS for tacking process complexity in my organization.	☐	☐	☒	☐	☐	

7 Open questionnaire

1. From what you have learned from presentations/documentation of MPMS and the demonstrated application of the component, what do you consider as the main positive and negative points?
   - Positive:
     - UI is good - good support
     - easy to configure from user perspective - it enables other
     - easy to understand how to use a program components to
     - easy to integrate work together
   - Negative:
     - MPMS doesn't work if info is not available
     - messages from PLC need to be translated so MPMS can understand it

2. Do you have any similar system(s) (e.g., MES) that cover(s) (part) of MPMS functionality?
   - If yes, please indicate what type of system(s) cover(s):
     - Process modeling: ……
     - Automated process execution: ……
     - Task delivery to humans and automated actors: ……
     - Exception handling: ……
     - Process monitoring: ……
   - No:……

another page 7.2

3. What functionality would you suggest that MPMS should cover, that currently the system misses?
   digital twin of what you are modeling to simulate, to see entire scenario and it's element

4. Which MPMS module do you consider difficult to use?
   I haven't seen the code

5. Any other comments/suggestions on the MPMS approach?
   no
   in UC1 there is a decision tree - because it's a simple scenario

(automated execution)
7.2 – Bosch software to manage AGV and SAP
those are similar but dedicated to specific
purpose – not flexible or general.

There isn't anything exactly like MPMS in [redacted]

Practitioner 2

[redacted]

May 20th
11⁰⁰

Manufacturing Process Management System (MPMS)

Evaluation form

1 Introduction

You are invited to participate in this survey with the purpose to evaluate the MPMS component. The survey is structured as follows:

1. Background information on your profile is collected, for assessing the suitability of the target participants.
2. Background information on MPMS is provided, to summarize its application purpose and functionality.
3. Evaluation of statements on perceived ease of use of MPMS.
4. Evaluation of statements on perceived usefulness of MPMS.
5. Evaluation of statements on intention to use MPMS.
6. Open questionnaire.

Note that this survey is not a test of knowledge or skills. There are no right or wrong answers. Please fill in the questions as realistically as possible.

The questionnaires consist of closed and open questions. Closed questions are answered by selecting one of the 5 available options. An example is given below.

		5-point likelihood scale				
	Statement	Strongly agree		Neutral		Strongly disagree
1	I like coffee better than tea.	☒	☐	☐	☐	☐
2	I do like sugar in my coffee.	☐	☐	☐	☒	☐

1.1 Privacy disclaimer

The data collected through this survey is processed anonymously and is under no circumstances used to trace answers back to individuals. The results are exclusively published in aggregated form, with the sole purpose of evaluating this research's proposed solution artifacts.

2 Participant's profile

- My current job title is: research eng
- My current job is best classified as (pick one):

☐ (Information) Systems Architect
☐ Developer
☑ Product/System Designer
☐ Operations Manager

M: at which lvl is exception handling done?

K: from devices, events (change in order)
it need to be captured in the process
e.g. if resource is not available
there is „go to another resource"
different lvl: task and process

M: Problem statement on slide 1.
Why BPMN methodology?

K: BPMN is popular (in manufacturing), widely accepted
for ISA 82...
to integrate business and production to use common notation

M: BPMN is the closest to bridge to manufacturing.

K: trying to bring BPMN from high to executable lvl

☒ Integrator
☒ Robotics expert
☐ Other (please specify): ....

Please indicate to which degree you agree with the following statements with respect to your background knowledge in Business Process Management (BPM):

	Statement	Strongly agree (5-point likelihood scale)		Neutral		Strongly disagree
1	I am familiar with BPM techniques and tools.	☐	☒	☐	☐	☐
2	I am interested in applying BPM techniques and tools.	☒	☐	☐	☐	☐
3	I consider myself expert in BPM.	☐	☒	☐	☐	☐
4	I am interested in process/workflow modeling.	☒	☐	☐	☐	☐
5	I am familiar with BPMN modeling.	☐	☒	☐	☐	☐
6	I consider myself expert in BPMN modeling.	☐	☒	☐	☐	☐

reading a lot ~~about~~ it about

3 MPMS background info

MPMS is based on Business Process Management (BPM) theories and tools. It is used for process modeling and automated orchestration of activities. It provides horizontal integration by providing cross-functional control (i.e., across production lines and work cells). It provides vertical control by triggering actions by humans and automated actors.

Please refer to the attached presentation on explanation on the modules of MPMS.

4 Perceived ease of use

Please indicate to which degree you agree with the following statements with respect to the perceived ease of use of the system.

	Statement	Strongly agree (5-point likelihood scale)		Neutral		Strongly disagree
	Perceived Ease of Use					
1.	Modeling processes with MPMS modeler is easy for me.	☐	☐	☐	☒	☐
2.	I find it difficult to provide definitions of involved entities (i.e., resources, tasks, location points) through MPMS.	☐	☐	☐	☒	☐
3.	I find that implementing any business logic (through coding an application project) is difficult for me.	☐	☒	☐	☐	☐

1. Not having a reference - too much freedom to do that. Some guidlines would be nice.
2. I didn't find too much difficulty
3. It was difficult (I hate Java) :) in python (external tasks) ~~[illegible]~~ was fine.

	Statement	Strongly agree		Neutral		Strongly disagree	
4.	The configuration and customization of MPMS is easy for me.	X					it was nice
5.	My interaction with the MPMS applications (Tasklist, Cockpit, Admin) is clear and understandable.	X					
6.	I find it takes a lot of effort to become skillful at using MPMS modules.		X				it takes some time
7.	I find MPMS rigid and inflexible to use.					X	
8.	Overall, I find MPMS easy to use.		X				

5 Perceived usefulness

Please indicate to which degree you agree with the following statements with respect to the perceived usefulness of the system.

	Statement	5-point likelihood scale					
		Strongly agree		Neutral		Strongly disagree	
	Perceived Usefulness						
1.	Using MPMS (Modeler) allows me to clearly represent (production) processes.		X				
2.	With MPMS (Modeler) it is possible to model complex production scenarios.			X			
3.	MPMS allows for process integration and automation that would be not possible (or difficult to achieve) without this system.		X				
4.	MPMS would enable higher productivity through process management. *your processes are more efficient*		X				it gives standardize way to have it that allow analysis
5.	With MPMS I get a clear overview of (production) processes.		X				limited to small use-case
6.	Overall, I find MPMS useful for tackling process complexity.			X			never use it

2. Use it for simple use-case. How would it be in more complex : too long. With exception handling etc it would be easier.

6 Intention to use

Please indicate to which degree you agree with the following statements with respect to the intention to use the system.

	Statement	5-point likelihood scale				
		Strongly agree		Neutral		Strongly disagree
	Intention to use					
1.	I would consider using the MPMS solutions for tacking process complexity in my organization.	X				
2.	I intend to use MPMS for tacking process complexity in my organization.	X				

2. May be used in Matteo phD thesis :)
exception thing is really interesting

7 Open questionnaire

1. From what you have learned from presentations/documentation of MPMS and the demonstrated application of the component, what do you consider as the main positive and negative points?
   - Positive:
     - drag and drop solution of BPMN
     - posibility to use the model for the orchestrate
     - exception handling lib.
     - ~~[illegible]~~ forum easy to debug, when building on open source solution
   - Negative:
     - suggestions how to modell
     - Java is an issue

2. Do you have any similar system(s) (e.g., MES) that cover(s) (part) of MPMS functionality?
   - If yes, please indicate what type of system(s) cover(s):
     - Process modeling: it can ≤ the Siemens MES
     - Automated process execution:
     - Task delivery to humans and automated actors:
     - Exception handling:
     - Process monitoring: no
   - No:........

notes on another page marked as 7. 2.

3. What functionality would you suggest that MPMS should cover, that currently the system misses?
   exception handling library / visualisation not relevant for SMEs
   library for communication
   check this not completed → to present something in the modeller (instead hand coding)

4. Which MPMS module do you consider difficult to use?
   core app backend in Java

5. Any other comments/suggestions on the MPMS approach?
   - nice to improve UI not limited to what Camunda provides
   - parametrize the Robot for complex things in modeller

Siemens

7.2. MES – bpmn flows are executed with extension (libraries)
it is just for automated agents
exception handling

In case of not having MES, MPMS could fill the gabs.
in siemens not the cased

Practitioner 3

[redacted]

May 31st
13:00

Manufacturing Process Management System (MPMS)

Evaluation form

1 Introduction

You are invited to participate in this survey with the purpose to evaluate the MPMS component. The survey is structured as follows:

1. Background information on your profile is collected, for assessing the suitability of the target participants.
2. Background information on MPMS is provided, to summarize its application purpose and functionality.
3. Evaluation of statements on perceived ease of use of MPMS.
4. Evaluation of statements on perceived usefulness of MPMS.
5. Evaluation of statements on intention to use MPMS.
6. Open questionnaire.

Note that this survey is not a test of knowledge or skills. There are no right or wrong answers. Please fill in the questions as realistically as possible.

The questionnaires consist of closed and open questions. Closed questions are answered by selecting one of the 5 available options. An example is given below.

		5-point likelihood scale				
	Statement	Strongly agree		Neutral		Strongly disagree
1	I like coffee better than tea.	☒	☐	☐	☐	☐
2	I do like sugar in my coffee.	☐	☐	☐	☒	☐

1.1 Privacy disclaimer

The data collected through this survey is processed anonymously and is under no circumstances used to trace answers back to individuals. The results are exclusively published in aggregated form, with the sole purpose of evaluating this research's proposed solution artifacts.

2 Participant's profile

- My current job title is: workflow architect
- My current job is best classified as (pick one):

☐ (Information) Systems Architect
☐ Developer
☐ Product/System Designer
☐ Operations Manager

☐ Integrator
☐ Robotics expert
☐ Other (please specify): ....

Please indicate to which degree you agree with the following statements with respect to your background knowledge in Business Process Management (BPM):

	Statement	5-point likelihood scale				
		Strongly agree		Neutral		Strongly disagree
1	I am familiar with BPM techniques and tools.	☐	☒	☐	☐	☐
2	I am interested in applying BPM techniques and tools.	☒	☐	☐	☐	☐
3	I consider myself expert in BPM.	☐	☐	☐	☒	☐
4	I am interested in process/workflow modeling.	☒	☐	☐	☐	☐
5	I am familiar with BPMN modeling.	☐	☒	☐	☐	☐
6	I consider myself expert in BPMN modeling.	☐	☐	☐	☒	☐

3 MPMS background info

MPMS is based on Business Process Management (BPM) theories and tools. It is used for process modeling and automated orchestration of activities. It provides horizontal integration by providing cross-functional control (i.e., across production lines and work cells). It provides vertical control by triggering actions by humans and automated actors.

Please refer to the attached presentation on explanation on the modules of MPMS.

4 Perceived ease of use

Please indicate to which degree you agree with the following statements with respect to the perceived ease of use of the system.

	Statement	5-point likelihood scale				
		Strongly agree		Neutral		Strongly disagree
	Perceived Ease of Use					
1.	Modeling processes with MPMS modeler is easy for me.	☐	☒	☐	☐	☐
2.	I find it difficult to provide definitions of involved entities (i.e., resources, tasks, location points) through MPMS.	☐	☐	☒	☐	☐
3.	I find that implementing any business logic (through coding an application project) is difficult for me.	☒	☐	☐	☐	☐

4.	The configuration and customization of MPMS is easy for me.	☐	☒	☐	☐	☐
5.	My interaction with the MPMS applications (Tasklist, Cockpit, Admin) is clear and understandable.	☒	☐	☐	☐	☐
6.	I find it takes a lot of effort to become skillful at using MPMS modules.	☐	☒	☐	☐	☐
7.	I find MPMS rigid and inflexible to use.	☐	☐	☐	☒	☐
8.	Overall, I find MPMS easy to use.	☐	☒	☐	☐	☐

5 Perceived usefulness

Please indicate to which degree you agree with the following statements with respect to the perceived usefulness of the system.

		5-point likelihood scale				
	Statement	Strongly agree		Neutral		Strongly disagree
	Perceived Usefulness					
1.	Using MPMS (Modeler) allows me to clearly represent (production) processes.	☒	☐	☐	☐	☐
2.	With MPMS (Modeler) it is possible to model complex production scenarios.	☐	☒	☐	☐	☐
3.	MPMS allows for process integration and automation that would be not possible (or difficult to achieve) without this system.	☒	☐	☐	☐	☐
4.	MPMS would enable higher productivity through process management.	☒	☐	☐	☐	☐
5.	With MPMS I get a clear overview of (production) processes.	☒	☐	☐	☐	☐
6.	Overall, I find MPMS useful for tackling process complexity.	☐	☒	☐	☐	☐

6 Intention to use

Please indicate to which degree you agree with the following statements with respect to the intention to use the system.

		5-point likelihood scale				
	Statement	Strongly agree		Neutral		Strongly disagree
	Intention to use					
1.	I would consider using the MPMS solutions for tacking process complexity in my organization.	☐	☒	☐	☐	☐
2.	I intend to use MPMS for tacking process complexity in my organization.	☐	☒	☐	☐	☐

7 Open questionnaire

1. From what you have learned from presentations/documentation of MPMS and the demonstrated application of the component, what do you consider as the main positive and negative points?
   - Positive:

     MPMS is nice for companies that starts with the production automation – cannot afford proprietary software
   - Negative:

     need for manual programming of each exception / order in that situation not possible to use MPMS as final solution (nice for prototyping the approach) offered to customers for them to construct the production ← not user-friendly if users is not IT expert / willing to code
2. Do you have any similar system(s) (e.g., MES) that cover(s) (part) of MPMS functionality?
   - If yes, please indicate what type of system(s) cover(s):
     - Process modeling: ……
     - Automated process execution: ……
     - Task delivery to humans and automated actors: unique to MPMS, not seen elsewhere
     - Exception handling: ……
     - Process monitoring: in current setup (on printing machines)
   - No:…….. there are no dashboard on general level, just local state of single machine
3. What functionality would you suggest that MPMS should cover, that currently the system misses?

   planning production to simulate it so the design (of production planning) can be tested and evaluated ← needed for optimization of production processes

   (K: may be included in analysis phase)
4. Which MPMS module do you consider difficult to use?

   ……
5. Any other comments/suggestions on the MPMS approach?

   ……

Practitioner 4

[redacted] - [redacted]

May 31st
13^{00}

Manufacturing Process Management System (MPMS)

Evaluation form

1 Introduction

You are invited to participate in this survey with the purpose to evaluate the MPMS component. The survey is structured as follows:

1. Background information on your profile is collected, for assessing the suitability of the target participants.
2. Background information on MPMS is provided, to summarize its application purpose and functionality.
3. Evaluation of statements on perceived ease of use of MPMS.
4. Evaluation of statements on perceived usefulness of MPMS.
5. Evaluation of statements on intention to use MPMS.
6. Open questionnaire.

Note that this survey is not a test of knowledge or skills. There are no right or wrong answers. Please fill in the questions as realistically as possible.

The questionnaires consist of closed and open questions. Closed questions are answered by selecting one of the 5 available options. An example is given below.

		5-point likelihood scale				
	Statement	Strongly agree		Neutral		Strongly disagree
1	I like coffee better than tea.	☒	☐	☐	☐	☐
2	I do like sugar in my coffee.	☐	☐	☐	☒	☐

1.1 Privacy disclaimer

The data collected through this survey is processed anonymously and is under no circumstances used to trace answers back to individuals. The results are exclusively published in aggregated form, with the sole purpose of evaluating this research's proposed solution artifacts.

2 Participant's profile

- My current job title is: work flow architect
- My current job is best classified as (pick one):

☐ (Information) Systems Architect
☐ Developer
☐ Product/System Designer
☐ Operations Manager

☐ Integrator
☐ Robotics expert
☐ Other (please specify): ....

Please indicate to which degree you agree with the following statements with respect to your background knowledge in Business Process Management (BPM):

	Statement	5-point likelihood scale				
		Strongly agree		Neutral		Strongly disagree
1	I am familiar with BPM techniques and tools.	☐	☒	☐	☐	☐
2	I am interested in applying BPM techniques and tools.	☒	☐	☐	☐	☐
3	I consider myself expert in BPM.	☐	☐	☐	☒	☐
4	I am interested in process/workflow modeling.	☒	☐	☐	☐	☐
5	I am familiar with BPMN modeling.	☐	☐	☐	☒	☐
6	I consider myself expert in BPMN modeling.	☐	☐	☐	☐	☒

3 MPMS background info

MPMS is based on Business Process Management (BPM) theories and tools. It is used for process modeling and automated orchestration of activities. It provides horizontal integration by providing cross-functional control (i.e., across production lines and work cells). It provides vertical control by triggering actions by humans and automated actors.

Please refer to the attached presentation on explanation on the modules of MPMS.

4 Perceived ease of use

Please indicate to which degree you agree with the following statements with respect to the perceived ease of use of the system.

	Statement	5-point likelihood scale				
		Strongly agree		Neutral		Strongly disagree
	Perceived Ease of Use					
1.	Modeling processes with MPMS modeler is easy for me.	☐	☒	☐	☐	☐
2.	I find it difficult to provide definitions of involved entities (i.e., resources, tasks, location points) through MPMS.	☐	☐	☒	☐	☐
3.	I find that implementing any business logic (through coding an application project) is difficult for me.	☒	☐	☐	☐	☐

4.	The configuration and customization of MPMS is easy for me.	☐	☒	☐	☐	☐
5.	My interaction with the MPMS applications (Tasklist, Cockpit, Admin) is clear and understandable.	☒	☐	☐	☐	☐
6.	I find it takes a lot of effort to become skillful at using MPMS modules.	☐	☒	☐	☐	☐
7.	I find MPMS rigid and inflexible to use.	☐	☐	☐	☒	☐
8.	Overall, I find MPMS easy to use.	☐	☒	☐	☐	☐

5 Perceived usefulness

Please indicate to which degree you agree with the following statements with respect to the perceived usefulness of the system.

		5-point likelihood scale				
		Strongly agree		Neutral		Strongly disagree
	Statement					
	Perceived Usefulness					
1.	Using MPMS (Modeler) allows me to clearly represent (production) processes.	☒	☐	☐	☐	☐
2.	With MPMS (Modeler) it is possible to model complex production scenarios.	☐	☒	☐	☐	☐
3.	MPMS allows for process integration and automation that would be not possible (or difficult to achieve) without this system.	☒	☐	☐	☐	☐
4.	MPMS would enable higher productivity through process management.	☒	☐	☐	☐	☐
5.	With MPMS I get a clear overview of (production) processes.	☒	☐	☐	☐	☐
6.	Overall, I find MPMS useful for tackling process complexity.	☐	☒	☐	☐	☐

6 Intention to use

Please indicate to which degree you agree with the following statements with respect to the intention to use the system.

		5-point likelihood scale				
		Strongly agree		Neutral		Strongly disagree
	Statement					
	Intention to use					
1.	I would consider using the MPMS solutions for tacking process complexity in my organization.	☐	☒	☐	☐	☐
2.	I intend to use MPMS for tacking process complexity in my organization.	☐	☒	☐	☐	☐

7 Open questionnaire

1. From what you have learned from presentations/documentation of MPMS and the demonstrated application of the component, what do you consider as the main positive and negative points?
   - Positive:
     ....we could achieve our goals with MPMS.........
     ....of testing new approach to production....
     ....management
     ....
   - Negative:
     ....
     .....a lot of details needed to model things
     ....
     ....method is very linear, perhaps cannot keep up the changing process/ flexible environment
2. Do you have any similar system(s) (e.g., MES) that cover(s) (part) of MPMS functionality?
   - If yes, please indicate what type of system(s) cover(s):
     - Process modeling: ....commercial systems..............
     - Automated process execution:
       ....
     - Task delivery to humans and automated actors:
       ....
     - Exception handling: ....
     - Process monitoring: ....
   - No:........

3. What functionality would you suggest that MPMS should cover, that currently the system misses?
   ....
   ....
   ....

4. Which MPMS module do you consider difficult to use?
   ....
   ....
   ....
   ....
   ....

5. Any other comments/suggestions on the MPMS approach?
   ....
   ....
   ....
   ....
   ....

question: new things wrt. digital transformation topics?

Kostas: data analysis, AI <- MPMS should be combined with those technologies

Practitioner 5

[redacted]

June 1st

11⁰⁰

Manufacturing Process Management System (MPMS)

Evaluation form

1 Introduction

You are invited to participate in this survey with the purpose to evaluate the MPMS component. The survey is structured as follows:

1. Background information on your profile is collected, for assessing the suitability of the target participants.
2. Background information on MPMS is provided, to summarize its application purpose and functionality.
3. Evaluation of statements on perceived ease of use of MPMS.
4. Evaluation of statements on perceived usefulness of MPMS.
5. Evaluation of statements on intention to use MPMS.
6. Open questionnaire.

Note that this survey is not a test of knowledge or skills. There are no right or wrong answers. Please fill in the questions as realistically as possible.

The questionnaires consist of closed and open questions. Closed questions are answered by selecting one of the 5 available options. An example is given below.

		5-point likelihood scale				
	Statement	Strongly agree		Neutral		Strongly disagree
1	I like coffee better than tea.	☒	☐	☐	☐	☐
2	I do like sugar in my coffee.	☐	☐	☐	☒	☐

1.1 Privacy disclaimer

The data collected through this survey is processed anonymously and is under no circumstances used to trace answers back to individuals. The results are exclusively published in aggregated form, with the sole purpose of evaluating this research's proposed solution artifacts.

2 Participant's profile

- My current job title is: ....managing...director
- My current job is best classified as (pick one):

☐ (Information) Systems Architect
☐ Developer
☐ Product/System Designer
☐ Operations Manager

☐ Integrator
☐ Robotics expert
☒ Other (please specify): ....general manager

Please indicate to which degree you agree with the following statements with respect to your background knowledge in Business Process Management (BPM):

	Statement	Strongly agree		Neutral		Strongly disagree
		5-point likelihood scale				
1	I am familiar with BPM techniques and tools.	☐	☒	☐	☐	☐
2	I am interested in applying BPM techniques and tools.	☒	☐	☐	☐	☐
3	I consider myself expert in BPM.	☐	☐	☐	☒	■
4	I am interested in process/workflow modeling.	■	☒	☐	☐	☐
5	I am familiar with BPMN modeling.	■	☒	☐	☐	☐
6	I consider myself expert in BPMN modeling.	☐	☐	☐	☒	☐

3 MPMS background info

MPMS is based on Business Process Management (BPM) theories and tools. It is used for process modeling and automated orchestration of activities. It provides horizontal integration by providing cross-functional control (i.e., across production lines and work cells). It provides vertical control by triggering actions by humans and automated actors.

Please refer to the attached presentation on explanation on the modules of MPMS.

4 Perceived ease of use

Please indicate to which degree you agree with the following statements with respect to the perceived ease of use of the system.

	Statement	Strongly agree		Neutral		Strongly disagree
		5-point likelihood scale				
	Perceived Ease of Use					
1.	Modeling processes with MPMS modeler is easy for me.	☐	☐	☐	☒	☐
2.	I find it difficult to provide definitions of involved entities (i.e., resources, tasks, location points) through MPMS.	☐	☐	☐	☐	☐
3.	I find that implementing any business logic (through coding an application project) is difficult for me.	☐	☐	☐	☒	☐

1) for the experts it is easier. We use it format to describe all our process sales marketing, quality, etc, mainly documentation not executed
don't know if BPMN is supported by new MES of TRI.
3) from organisasation point of view not so difficult

4.	The configuration and customization of MPMS is easy for me.	☐	☒	☐	☐	☐
5.	My interaction with the MPMS applications (Tasklist, Cockpit, Admin) is clear and understandable.	☐	☒	☐	☐	☐
6.	I find it takes a lot of effort to become skillful at using MPMS modules.	☐	☒	☐	☐	☐
7.	I find MPMS rigid and inflexible to use.	☐	☐	☐	☒	☐
8.	Overall, I find MPMS easy to use.	☐	☒	☐	☐	☐

need so time to get used to modelling

7. if organisation is too small it has no capacity to pick up the ball

5 Perceived usefulness

Please indicate to which degree you agree with the following statements with respect to the perceived usefulness of the system.

	Statement	5-point likelihood scale: Strongly agree		Neutral		Strongly disagree
	Perceived Usefulness					
1.	Using MPMS (Modeler) allows me to clearly represent (production) processes.	☐	☒	☐	☐	☐
2.	With MPMS (Modeler) it is possible to model complex production scenarios.	☒	☐	☐	☐	☐
3.	MPMS allows for process integration and automation that would be not possible (or difficult to achieve) without this system.	☐	☐	☒	☐	☐
4.	MPMS would enable higher productivity through process management.	☐	☒	☐	☐	☐
5.	With MPMS I get a clear overview of (production) processes.	☐	☒	☐	☐	☐
6.	Overall, I find MPMS useful for tackling process complexity.	☐	☒	☐	☐	☐

not skillful enough to answer

tricky one

if there is no support, small company won't go all the way

6 Intention to use

Please indicate to which degree you agree with the following statements with respect to the intention to use the system.

	Statement	5-point likelihood scale: Strongly agree		Neutral		Strongly disagree
	Intention to use					
1.	I would consider using the MPMS solutions for tacking process complexity in my organization.	☒	☐	☐	☐	☐
2.	I intend to use MPMS for tacking process complexity in my organization.	☐	☒	☐	☐	☐

partly, no exec.

7 Open questionnaire

1. From what you have learned from presentations/documentation of MPMS and the demonstrated application of the component, what do you consider as the main positive and negative points?
   - Positive:
     end-to-end process perspective, safty
     use it for explanation reasons - to train operators
     …………………………………………………………………………………
     …………………………………………………………………………………
   - Negative:
     this modelling needs continuity; 5-6 TRL too low lvl to implement it
     …………………………………………………………………………………
     …………………………………………………………………………………
     …………………………………………………………………………………

2. Do you have any similar system(s) (e.g., MES) that cover(s) (part) of MPMS functionality?
   - If yes, please indicate what type of system(s) cover(s):
     - Process modeling: ………………………………………………………
     - Automated process execution: ………………………………………………………
     - Task delivery to humans and automated actors: ………………………………………………………
     - Exception handling: ………………………………………………………
     - Process monitoring: ………………………………………………………
   - No:……..

3. What functionality would you suggest that MPMS should cover, that currently the system misses?
   …………………………………………………………………………………
   …………………………………………………………………………………
   …………………………………………………………………………………

4. Which MPMS module do you consider difficult to use?
   …………………………………………………………………………………
   …………………………………………………………………………………
   …………………………………………………………………………………
   …………………………………………………………………………………
   …………………………………………………………………………………

5. Any other comments/suggestions on the MPMS approach?
   …………………………………………………………………………………
   …………………………………………………………………………………
   …………………………………………………………………………………
   …………………………………………………………………………………
   …………………………………………………………………………………

2. Currenty in TRI info flow MES -> WC; other way WC -> MES very limited

Practitioner 6

[redacted] - [redacted] [redacted]

June 6th
10^{00}

Manufacturing Process Management System (MPMS)

Evaluation form

1 Introduction

You are invited to participate in this survey with the purpose to evaluate the MPMS component. The survey is structured as follows:

1. Background information on your profile is collected, for assessing the suitability of the target participants.
2. Background information on MPMS is provided, to summarize its application purpose and functionality.
3. Evaluation of statements on perceived ease of use of MPMS.
4. Evaluation of statements on perceived usefulness of MPMS.
5. Evaluation of statements on intention to use MPMS.
6. Open questionnaire.

Note that this survey is not a test of knowledge or skills. There are no right or wrong answers. Please fill in the questions as realistically as possible.

The questionnaires consist of closed and open questions. Closed questions are answered by selecting one of the 5 available options. An example is given below.

		5-point likelihood scale				
	Statement	Strongly agree		Neutral		Strongly disagree
1	I like coffee better than tea.	☒	☐	☐	☐	☐
2	I do like sugar in my coffee.	☐	☐	☐	☒	☐

1.1 Privacy disclaimer

The data collected through this survey is processed anonymously and is under no circumstances used to trace answers back to individuals. The results are exclusively published in aggregated form, with the sole purpose of evaluating this research's proposed solution artifacts.

2 Participant's profile

- My current job title is: ....CO / head of products..........
- My current job is best classified as (pick one):

☐ (Information) Systems Architect
☐ Developer
☒ Product/System Designer
☐ Operations Manager

☐ Integrator
☐ Robotics expert
☐ Other (please specify): ……………………………………

Please indicate to which degree you agree with the following statements with respect to your background knowledge in Business Process Management (BPM):

	Statement	5-point likelihood scale: Strongly agree		Neutral		Strongly disagree
1	I am familiar with BPM techniques and tools.	☐	☒	☐	☐	☐
2	I am interested in applying BPM techniques and tools.	☐	☒	☐	☐	☐
3	I consider myself expert in BPM.	☐	☒	☒	☐	☐
4	I am interested in process/workflow modeling.	☐	☒	☐	☐	☐
5	I am familiar with BPMN modeling.	☐	☒	☐	☐	☐
6	I consider myself expert in BPMN modeling.	☐	☐	☒	☐	☐

3 MPMS background info

MPMS is based on Business Process Management (BPM) theories and tools. It is used for process modeling and automated orchestration of activities. It provides horizontal integration by providing cross-functional control (i.e., across production lines and work cells). It provides vertical control by triggering actions by humans and automated actors.

Please refer to the attached presentation on explanation on the modules of MPMS.

4 Perceived ease of use

Please indicate to which degree you agree with the following statements with respect to the perceived ease of use of the system.

	Statement	5-point likelihood scale: Strongly agree		Neutral		Strongly disagree
	Perceived Ease of Use					
1.	Modeling processes with MPMS modeler is easy for me.	☐	☒	☐	☐	☐
2.	I find it difficult to provide definitions of involved entities (i.e., resources, tasks, location points) through MPMS.	☐	☐	☒	☐	☐
3.	I find that implementing any business logic (through coding an application project) is difficult for me.	☐	☐	☐	☒	☐

It's easy

2. struggle with MPMS and used Camunda instead
the definition of entities is not what Dobek would like to follow
3 it may be difficult in Java, we write in own language
and use REST-API

7. inflexible is not the right word. there are missing some parts but it has noting to do with flexibility

4.	The configuration and customization of MPMS is easy for me.	☐	☐	☐	☒	☐
5.	My interaction with the MPMS applications (Tasklist, Cockpit, Admin) is clear and understandable.	☐	☒	☐	☐	☐
6.	I find it takes a lot of effort to become skillful at using MPMS modules.	☐	☐	☐	☒	☐
7.	I find MPMS rigid and inflexible to use.	☐	☐	☐	☒	☐
8.	Overall, I find MPMS easy to use.	☐	☒	☐	☐	☐

in general the concept it's easy to use

5 Perceived usefulness

Please indicate to which degree you agree with the following statements with respect to the perceived usefulness of the system.

3. MPMS follows orchiestration paterns
I've seen scenarios without bpmn and it worked in terms of automation and integration simply, different approach, you can achieve that in a different constructs; shared data models, distributed data models etc
from some factories point of view it's better to treat MPMS as another service and not to keep data only in MPMS as it depends on the other systems and devices, system dynamics etc

to keep separation on right lvl

	Statement	Strongly agree		Neutral		Strongly disagree
	Perceived Usefulness					
1.	Using MPMS (Modeler) allows me to clearly represent (production) processes.	☐	☒	☐	☐	☐
2.	With MPMS (Modeler) it is possible to model complex production scenarios.	☐	☒	☐	☐	☐
3.	MPMS allows for process integration and automation that would be not possible (or difficult to achieve) without this system.	☐	☐	☐	☒	☐
4.	MPMS would enable higher productivity through process management.	☐	☑	☐	☐	☐
5.	With MPMS I get a clear overview of (production) processes.	☐	☒	☐	☐	☐
6.	Overall, I find MPMS useful for tackling process complexity.	☐	☒	☐	☐	☐

(5-point likelihood scale)

from the presentation

it depends on many factors; practical usage is customare-depend

if you put enough effort then it's easy

from presentation it's perfect to describe reality

6 Intention to use

Please indicate to which degree you agree with the following statements with respect to the intention to use the system.

	Statement	Strongly agree		Neutral		Strongly disagree
	Intention to use					
1.	I would consider using the MPMS solutions for tacking process complexity in my organization.	☐	☐	☐	☒	☐
2.	I intend to use MPMS for tacking process complexity in my organization.	☐	☒	☐	☐	☐

(5-point likelihood scale)

maintanance cost

not for tackling process

1. if you have to introduce it's not time-effective
2. when you have big variations among customers, data agnostic

7 Open questionnaire

1. From what you have learned from presentations/documentation of MPMS and the demonstrated application of the component, what do you consider as the main positive and negative points?
   - Positive:
     you can perfectly structure the reality
     concept vision is interesting
     …………………………………………………………………………………………
     …………………………………………………………………………………………
   - Negative:
     if you are missing the workflow it's bad, need to put a lot of effort (for SME may not be the right thing), building image itself (deployment issues, → depends on Java etc)
     you need maintenance policy if writing some code on top of MPMS
     should be infrastructure agnostic, lack of prediction unt stability
     clear description of what component may deliver - responsibility
     (not research issues just comercialisation comments)
2. Do you have any similar system(s) (e.g., MES) that cover(s) (part) of MPMS functionality?
   - If yes, please indicate what type of system(s) cover(s):
     - Process modeling: ………………………………………………………………
     - Automated process execution: …………………………………………………………………
     - Task delivery to humans and automated actors: …………………………………………………………………
     - Exception handling: ………………………………………………………………
     - Process monitoring: ………………………………………………………………
   - No:……..

   Manufacturing Operation System (Management) partially covers those functionalities but with different flavour / focus
   software of HPDV, PCI Siemens, GE
3. What functionality would you suggest that MPMS should cover, that currently the system misses?
   ………………………………………………………………………………………………
   ………………………………………………………………………………………………
   ………………………………………………………………………………………………
4. Which MPMS module do you consider difficult to use?
   ………………………………………………………………………………………………
   ………………………………………………………………………………………………
   ………………………………………………………………………………………………
   ………………………………………………………………………………………………
   ………………………………………………………………………………………………
5. Any other comments/suggestions on the MPMS approach?
   ………………………………………………………………………………………………
   ………………………………………………………………………………………………
   ………………………………………………………………………………………………
   ………………………………………………………………………………………………
   ………………………………………………………………………………………………

About the author

Konstantinos Traganos was born on the 10th of December 1984, in Lamia, Greece. After finishing his secondary education at the General High School of Stylida, he started his 5-years undergraduate studies in Electrical & Computer Engineering at the National Technical University of Athens (NTUA), Greece. He obtained his Diploma in 2007 with specialization in networks and telecommunications. After fulfilling his obligatory military services, he worked for several years as a software developer in Athens, Greece. In 2012, he moved to Eindhoven, the Netherlands, to pursue a Master of Science (MSc) degree in Business Information Systems at the Technical University of Eindhoven (TUE). During his postgraduate studies, he also worked as a student tutor for bachelor students in the discipline of information systems, offered by the Computer Science department of the university. He obtained his MSc in 2014 with specialization in business process management (BPM) and information systems architecture. Upon completion of the master's degree, he worked for half a year as a research assistant in the Information Systems (IS) research group of the department of Industrial Engineering and Innovation Sciences at TUE, focusing on the design of reference architectures of intelligent transport systems and the design of business models in a service-dominant context in smart mobility domain, within the DITCM project. Afterwards, he moved to Luxembourg City, Luxembourg, to work as a software engineer.

Konstantinos returned to IS group in October 2016 to work as a business information systems engineer for the Horizon 2020 research and innovation HORSE project. His main tasks were requirement analysis of smart manufacturing processes, the development of a Manufacturing Process Management System (MPMS) for process orchestration in factories and the design of a reference architecture for cyber-physical systems. Since 2018, he started his PhD research on applying BPM concepts and tooling in smart manufacturing. His research on this topic is communicated through this dissertation and several publications pointed out in the dissertation. Apart from HORSE project, his research has also been carried out as part of the EIT OEDIPUS and SHOP4CF projects and under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems. During the past six years, Konstantinos has been involved in the supervision of several BSc end projects and MSc projects in the discipline of information systems. He has also engaged in teaching activities in the courses of BPM, business information systems architecture, and business analysis for IT systems.

His research interests include application of all phases of BPM in enterprises in various domains, design of information systems architectures for supporting organizations' activities, software and process requirements engineering, and software development methodologies. Overall, he is intrigued by challenges of putting structure in complex systems.

SIKS Dissertation Series

====
2011
====

2011-01 Botond Cseke (RUN)
Variational Algorithms for Bayesian Inference in Latent Gaussian Models

2011-02 Nick Tinnemeier(UU)
Organizing Agent Organizations. Syntax and Operational Semantics of an Organization-Oriented Programming Language

2011-03 Jan Martijn van der Werf (TUE)
Compositional Design and Verification of Component-Based Information Systems

2011-04 Hado van Hasselt (UU)
Insights in Reinforcement Learning; Formal analysis and empirical evaluation of temporal-difference learning algorithms

2011-05 Base van der Raadt (VU)
Enterprise Architecture Coming of Age - Increasing the Performance of an Emerging Discipline

2011-06 Yiwen Wang (TUE)
Semantically-Enhanced Recommendations in Cultural Heritage

2011-07 Yujia Cao (UT)
Multimodal Information Presentation for High Load Human Computer Interaction

2011-08 Nieske Vergunst (UU)
BDI-based Generation of Robust Task-Oriented Dialogues

2011-09 Tim de Jong (OU)
Contextualised Mobile Media for Learning

2011-10 Bart Bogaert (UvT)
Cloud Content Contention

2011-11 Dhaval Vyas (UT)
Designing for Awareness: An Experience-focused HCI Perspective

2011-12 Carmen Bratosin (TUE)
Grid Architecture for Distributed Process Mining

2011-13 Xiaoyu Mao (UvT)
Airport under Control. Multiagent Scheduling for Airport Ground Handling

2011-14 Milan Lovric (EUR)
Behavioral Finance and Agent-Based Artificial Markets

2011-15 Marijn Koolen (UvA)
The Meaning of Structure: the Value of Link Evidence for Information Retrieval

2011-16 Maarten Schadd (UM)
Selective Search in Games of Different Complexity

2011-17 Jiyin He (UVA)
Exploring Topic Structure: Coherence, Diversity and Relatedness

2011-18 Mark Ponsen (UM)
Strategic Decision-Making in complex games

2011-19 Ellen Rusman (OU)
The Mind ' s Eye on Personal Profiles

2011-20 Qing Gu (VU)
Guiding service-oriented software engineering - A view-based approach

2011-21 Linda Terlouw (TUD)
Modularization and Specification of Service-Oriented Systems

2011-22 Junte Zhang (UVA)
System Evaluation of Archival Description and Access

2011-23 Wouter Weerkamp (UVA)
Finding People and their Utterances in Social Media

2011-24 Herwin van Welbergen (UT)

Behavior Generation for Interpersonal Coordination with Virtual Humans On Specifying, Scheduling and Realizing Multimodal Virtual Human Behavior

2011-25 Syed Waqar ul Qounain Jaffry (VU)
Analysis and Validation of Models for Trust Dynamics

2011-26 Matthijs Aart Pontier (VU)
Virtual Agents for Human Communication - Emotion Regulation and Involvement-Distance Trade-Offs in Embodied Conversational Agents and Robots

2011-27 Aniel Bhulai (VU)
Dynamic website optimization through autonomous management of design patterns

2011-28 Rianne Kaptein (UVA)
Effective Focused Retrieval by Exploiting Query Context and Document Structure

2011-29 Faisal Kamiran (TUE)
Discrimination-aware Classification

2011-30 Egon van den Broek (UT)
Affective Signal Processing (ASP): Unraveling the mystery of emotions

2011-31 Ludo Waltman (EUR)
Computational and Game-Theoretic Approaches for Modeling Bounded Rationality

2011-32 Nees-Jan van Eck (EUR)
Methodological Advances in Bibliometric Mapping of Science

2011-33 Tom van der Weide (UU)
Arguing to Motivate Decisions

2011-34 Paolo Turrini (UU)
Strategic Reasoning in Interdependence: Logical and Game-theoretical Investigations

2011-35 Maaike Harbers (UU)
Explaining Agent Behavior in Virtual Training

2011-36 Erik van der Spek (UU)
Experiments in serious game design: a cognitive approach

2011-37 Adriana Burlutiu (RUN)
Machine Learning for Pairwise Data, Applications for Preference Learning and Supervised Network Inference

2011-38 Nyree Lemmens (UM)
Bee-inspired Distributed Optimization

2011-39 Joost Westra (UU)
Organizing Adaptation using Agents in Serious Games

2011-40 Viktor Clerc (VU)
Architectural Knowledge Management in Global Software Development

2011-41 Luan Ibraimi (UT)
Cryptographically Enforced Distributed Data Access Control

2011-42 Michal Sindlar (UU)
Explaining Behavior through Mental State Attribution

2011-43 Henk van der Schuur (UU)
Process Improvement through Software Operation Knowledge

2011-44 Boris Reuderink (UT)
Robust Brain-Computer Interfaces

2011-45 Herman Stehouwer (UvT)
Statistical Language Models for Alternative Sequence Selection

2011-46 Beibei Hu (TUD)
Towards Contextualized Information Delivery: A Rule-based Architecture for the Domain of Mobile Police Work

2011-47 Azizi Bin Ab Aziz (VU)
Exploring Computational Models for Intelligent Support of Persons with Depression

2011-48 Mark Ter Maat (UT)
Response Selection and Turn-taking for a Sensitive Artificial Listening Agent

2011-49 Andreea Niculescu (UT)
Conversational interfaces for task-oriented spoken dialogues: design aspects influencing interaction quality

====

2012

====

2012-01	Terry Kakeeto (UvT) Relationship Marketing for SMEs in Uganda
2012-02	Muhammad Umair (VU) Adaptivity, emotion, and Rationality in Human and Ambient Agent Models
2012-03	Adam Vanya (VU) Supporting Architecture Evolution by Mining Software Repositories
2012-04	Jurriaan Souer (UU) Development of Content Management System-based Web Applications
2012-05	Marijn Plomp (UU) Maturing Interorganisational Information Systems
2012-06	Wolfgang Reinhardt (OU) Awareness Support for Knowledge Workers in Research Networks
2012-07	Rianne van Lambalgen (VU) When the Going Gets Tough: Exploring Agent-based Models of Human Performance under Demanding Conditions
2012-08	Gerben de Vries (UVA) Kernel Methods for Vessel Trajectories
2012-09	Ricardo Neisse (UT) Trust and Privacy Management Support for Context-Aware Service Platforms
2012-10	David Smits (TUE) Towards a Generic Distributed Adaptive Hypermedia Environment
2012-11	J.C.B. Rantham Prabhakara (TUE) Process Mining in the Large: Preprocessing, Discovery, and Diagnostics
2012-12	Kees van der Sluijs (TUE) Model Driven Design and Data Integration in Semantic Web Information Systems
2012-13	Suleman Shahid (UvT) Fun and Face: Exploring non-verbal expressions of emotion during playful interactions
2012-14	Evgeny Knutov (TUE) Generic Adaptation Framework for Unifying Adaptive Web-based Systems
2012-15	Natalie van der Wal (VU) Social Agents. Agent-Based Modelling of Integrated Internal and Social Dynamics of Cognitive and Affective Processes
2012-16	Fiemke Both (VU) Helping people by understanding them - Ambient Agents supporting task execution and depression treatment
2012-17	Amal Elgammal (UvT) Towards a Comprehensive Framework for Business Process Compliance
2012-18	Eltjo Poort (VU) Improving Solution Architecting Practices
2012-19	Helen Schonenberg (TUE) What's Next? Operational Support for Business Process Execution
2012-20	Ali Bahramisharif (RUN) Covert Visual Spatial Attention, a Robust Paradigm for Brain-Computer Interfacing
2012-21	Roberto Cornacchia (TUD) Querying Sparse Matrices for Information Retrieval
2012-22	Thijs Vis (UvT) Intelligence, politie en veiligheidsdienst: verenigbare grootheden?
2012-23	Christian Muehl (UT) Toward Affective Brain-Computer Interfaces: Exploring the Neurophysiology of Affect during Human Media Interaction
2012-24	Laurens van der Werff (UT) Evaluation of Noisy Transcripts for Spoken Document Retrieval
2012-25	Silja Eckartz (UT)

Managing the Business Case Development in Inter-Organizational IT Projects: A Methodology and its Application

2012-26 Emile de Maat (UVA)
Making Sense of Legal Text

2012-27 Hayrettin Gurkok (UT)
Mind the Sheep! User Experience Evaluation & Brain-Computer Interface Games

2012-28 Nancy Pascall (UvT)
Engendering Technology Empowering Women

2012-29 Almer Tigelaar (UT)
Peer-to-Peer Information Retrieval

2012-30 Alina Pommeranz (TUD)
Designing Human-Centered Systems for Reflective Decision Making

2012-31 Emily Bagarukayo (RUN)
A Learning by Construction Approach for Higher Order Cognitive Skills Improvement, Building Capacity and Infrastructure

2012-32 Wietske Visser (TUD)
Qualitative multi-criteria preference representation and reasoning

2012-33 Rory Sie (OUN)
Coalitions in Cooperation Networks (COCOON)

2012-34 Pavol Jancura (RUN)
Evolutionary analysis in PPI networks and applications

2012-35 Evert Haasdijk (VU)
Never Too Old To Learn -- On-line Evolution of Controllers in Swarm- and Modular Robotics

2012-36 Denis Ssebugwawo (RUN)
Analysis and Evaluation of Collaborative Modeling Processes

2012-37 Agnes Nakakawa (RUN)
A Collaboration Process for Enterprise Architecture Creation

2012-38 Selmar Smit (VU)
Parameter Tuning and Scientific Testing in Evolutionary Algorithms

2012-39 Hassan Fatemi (UT)
Risk-aware design of value and coordination networks

2012-40 Agus Gunawan (UvT)
Information Access for SMEs in Indonesia

2012-41 Sebastian Kelle (OU)
Game Design Patterns for Learning

2012-42 Dominique Verpoorten (OU)
Reflection Amplifiers in self-regulated Learning

2012-43 Withdrawn

2012-44 Anna Tordai (VU)
On Combining Alignment Techniques

2012-45 Benedikt Kratz (UvT)
A Model and Language for Business-aware Transactions

2012-46 Simon Carter (UVA)
Exploration and Exploitation of Multilingual Data for Statistical Machine Translation

2012-47 Manos Tsagkias (UVA)
Mining Social Media: Tracking Content and Predicting Behavior

2012-48 Jorn Bakker (TUE)
Handling Abrupt Changes in Evolving Time-series Data

2012-49 Michael Kaisers (UM)
Learning against Learning - Evolutionary dynamics of reinforcement learning algorithms in strategic interactions

2012-50 Steven van Kervel (TUD)
Ontologogy driven Enterprise Information Systems Engineering

2012-51 Jeroen de Jong (TUD)
Heuristics in Dynamic Sceduling; a practical framework with a case study in elevator dispatching

====
2013
====

2013-01 Viorel Milea (EUR)
News Analytics for Financial Decision Support
2013-02 Erietta Liarou (CWI)
MonetDB/DataCell: Leveraging the Column-store Database Technology for Efficient and Scalable Stream Processing
2013-03 Szymon Klarman (VU)
Reasoning with Contexts in Description Logics
2013-04 Chetan Yadati(TUD)
Coordinating autonomous planning and scheduling
2013-05 Dulce Pumareja (UT)
Groupware Requirements Evolutions Patterns
2013-06 Romulo Goncalves (CWI)
The Data Cyclotron: Juggling Data and Queries for a Data Warehouse Audience
2013-07 Giel van Lankveld (UvT)
Quantifying Individual Player Differences
2013-08 Robbert-Jan Merk (VU)
Making enemies: cognitive modeling for opponent agents in fighter pilot simulators
2013-09 Fabio Gori (RUN)
Metagenomic Data Analysis: Computational Methods and Applications
2013-10 Jeewanie Jayasinghe Arachchige(UvT)
A Unified Modeling Framework for Service Design
2013-11 Evangelos Pournaras (TUD)
Multi-level Reconfigurable Self-organization in Overlay Services
2013-12 Marian Razavian (VU)
Knowledge-driven Migration to Services
2013-13 Mohammad Safiri (UT)
Service Tailoring: User-centric creation of integrated IT-based homecare services to support independent living of elderly
2013-14 Jafar Tanha (UVA)
Ensemble Approaches to Semi-Supervised Learning Learning
2013-15 Daniel Hennes (UM)
Multiagent Learning - Dynamic Games and Applications
2013-16 Eric Kok (UU)
Exploring the practical benefits of argumentation in multi-agent deliberation
2013-17 Koen Kok (VU)
The PowerMatcher: Smart Coordination for the Smart Electricity Grid
2013-18 Jeroen Janssens (UvT)
Outlier Selection and One-Class Classification
2013-19 Renze Steenhuizen (TUD)
Coordinated Multi-Agent Planning and Scheduling
2013-20 Katja Hofmann (UvA)
Fast and Reliable Online Learning to Rank for Information Retrieval
2013-21 Sander Wubben (UvT)
Text-to-text generation by monolingual machine translation
2013-22 Tom Claassen (RUN)
Causal Discovery and Logic
2013-23 Patricio de Alencar Silva (UvT)
Value Activity Monitoring
2013-24 Haitham Bou Ammar (UM)
Automated Transfer in Reinforcement Learning
2013-25 Agnieszka Anna Latoszek-Berendsen (UM)
Intention-based Decision Support. A new way of representing and implementing clinical guidelines in a Decision Support System

2013-26 Alireza Zarghami (UT)
Architectural Support for Dynamic Homecare Service Provisioning
2013-27 Mohammad Huq (UT)
Inference-based Framework Managing Data Provenance
2013-28 Frans van der Sluis (UT)
When Complexity becomes Interesting: An Inquiry into the Information eXperience
2013-29 Iwan de Kok (UT)
Listening Heads
2013-30 Joyce Nakatumba (TUE)
Resource-Aware Business Process Management: Analysis and Support
2013-31 Dinh Khoa Nguyen (UvT)
Blueprint Model and Language for Engineering Cloud Applications
2013-32 Kamakshi Rajagopal (OUN)
Networking For Learning; The role of Networking in a Lifelong Learner's Professional Development
2013-33 Qi Gao (TUD)
User Modeling and Personalization in the Microblogging Sphere
2013-34 Kien Tjin-Kam-Jet (UT)
Distributed Deep Web Search
2013-35 Abdallah El Ali (UvA)
Minimal Mobile Human Computer Interaction
Promotor: Prof. dr. L. Hardman (CWI/UVA)
2013-36 Than Lam Hoang (TUe)
Pattern Mining in Data Streams
2013-37 Dirk Börner (OUN)
Ambient Learning Displays
2013-38 Eelco den Heijer (VU)
Autonomous Evolutionary Art
2013-39 Joop de Jong (TUD)
A Method for Enterprise Ontology based Design of Enterprise Information Systems
2013-40 Pim Nijssen (UM)
Monte-Carlo Tree Search for Multi-Player Games
2013-41 Jochem Liem (UVA)
Supporting the Conceptual Modelling of Dynamic Systems: A Knowledge Engineering Perspective on Qualitative Reasoning
2013-42 Léon Planken (TUD)
Algorithms for Simple Temporal Reasoning
2013-43 Marc Bron (UVA)
Exploration and Contextualization through Interaction and Concepts

====
2014
====

2014-01 Nicola Barile (UU)
Studies in Learning Monotone Models from Data
2014-02 Fiona Tuliyano (RUN)
Combining System Dynamics with a Domain Modeling Method
2014-03 Sergio Raul Duarte Torres (UT)
Information Retrieval for Children: Search Behavior and Solutions
2014-04 Hanna Jochmann-Mannak (UT)
Websites for children: search strategies and interface design - Three studies on children's search performance and evaluation
2014-05 Jurriaan van Reijsen (UU)
Knowledge Perspectives on Advancing Dynamic Capability
2014-06 Damian Tamburri (VU)
Supporting Networked Software Development

2014-07 Arya Adriansyah (TUE)
Aligning Observed and Modeled Behavior
2014-08 Samur Araujo (TUD)
Data Integration over Distributed and Heterogeneous Data Endpoints
2014-09 Philip Jackson (UvT)
Toward Human-Level Artificial Intelligence: Representation and Computation of Meaning in Natural Language
2014-10 Ivan Salvador Razo Zapata (VU)
Service Value Networks
2014-11 Janneke van der Zwaan (TUD)
An Empathic Virtual Buddy for Social Support
2014-12 Willem van Willigen (VU)
Look Ma, No Hands: Aspects of Autonomous Vehicle Control
2014-13 Arlette van Wissen (VU)
Agent-Based Support for Behavior Change: Models and Applications in Health and Safety Domains
2014-14 Yangyang Shi (TUD)
Language Models With Meta-information
2014-15 Natalya Mogles (VU)
Agent-Based Analysis and Support of Human Functioning in Complex Socio-Technical Systems: Applications in Safety and Healthcare
2014-16 Krystyna Milian (VU)
Supporting trial recruitment and design by automatically interpreting eligibility criteria
2014-17 Kathrin Dentler (VU)
Computing healthcare quality indicators automatically: Secondary Use of Patient Data and Semantic Interoperability
2014-18 Mattijs Ghijsen (UVA)
Methods and Models for the Design and Study of Dynamic Agent Organizations
2014-19 Vinicius Ramos (TUE)
Adaptive Hypermedia Courses: Qualitative and Quantitative Evaluation and Tool Support
2014-20 Mena Habib (UT)
Named Entity Extraction and Disambiguation for Informal Text: The Missing Link
2014-21 Kassidy Clark (TUD)
Negotiation and Monitoring in Open Environments
2014-22 Marieke Peeters (UU)
Personalized Educational Games - Developing agent-supported scenario-based training
2014-23 Eleftherios Sidirourgos (UvA/CWI)
Space Efficient Indexes for the Big Data Era
2014-24 Davide Ceolin (VU)
Trusting Semi-structured Web Data
2014-25 Martijn Lappenschaar (RUN)
New network models for the analysis of disease interaction
2014-26 Tim Baarslag (TUD)
What to Bid and When to Stop
2014-27 Rui Jorge Almeida (EUR)
Conditional Density Models Integrating Fuzzy and Probabilistic Representations of Uncertainty
2014-28 Anna Chmielowiec (VU)
Decentralized k-Clique Matching
2014-29 Jaap Kabbedijk (UU)
Variability in Multi-Tenant Enterprise Software
2014-30 Peter de Cock (UvT)
Anticipating Criminal Behaviour
2014-31 Leo van Moergestel (UU)
Agent Technology in Agile Multiparallel Manufacturing and Product Support
2014-32 Naser Ayat (UvA)
On Entity Resolution in Probabilistic Data
2014-33 Tesfa Tegegne (RUN)
Service Discovery in eHealth

2014-34 Christina Manteli (VU)
The Effect of Governance in Global Software Development: Analyzing Transactive Memory Systems
2014-35 Joost van Ooijen (UU)
Cognitive Agents in Virtual Worlds: A Middleware Design Approach
2014-36 Joos Buijs (TUE)
Flexible Evolutionary Algorithms for Mining Structured Process Models
2014-37 Maral Dadvar (UT)
Experts and Machines United Against Cyberbullying
2014-38 Danny Plass-Oude Bos (UT)
Making brain-computer interfaces better: improving usability through post-processing
2014-39 Jasmina Maric (UvT)
Web Communities, Immigration, and Social Capital
2014-40 Walter Omona (RUN)
A Framework for Knowledge Management Using ICT in Higher Education
2014-41 Frederic Hogenboom (EUR)
Automated Detection of Financial Events in News Text
2014-42 Carsten Eijckhof (CWI/TUD)
Contextual Multidimensional Relevance Models
2014-43 Kevin Vlaanderen (UU)
Supporting Process Improvement using Method Increments
2014-44 Paulien Meesters (UvT)
Intelligent Blauw. Met als ondertitel: Intelligence-gestuurde politiezorg in gebiedsgebonden eenheden
2014-45 Birgit Schmitz (OUN)
Mobile Games for Learning: A Pattern-Based Approach
2014-46 Ke Tao (TUD)
Social Web Data Analytics: Relevance, Redundancy, Diversity
2014-47 Shangsong Liang (UVA)
Fusion and Diversification in Information Retrieval

====
2015
====

2015-01 Niels Netten (UvA)
Machine Learning for Relevance of Information in Crisis Response
2015-02 Faiza Bukhsh (UvT)
Smart auditing: Innovative Compliance Checking in Customs Controls
2015-03 Twan van Laarhoven (RUN)
Machine learning for network data
2015-04 Howard Spoelstra (OUN)
Collaborations in Open Learning Environments
2015-05 Christoph Bösch (UT)
Cryptographically Enforced Search Pattern Hiding
2015-06 Farideh Heidari (TUD)
Business Process Quality Computation - Computing Non-Functional Requirements to Improve Business Processes
2015-07 Maria-Hendrike Peetz(UvA)
Time-Aware Online Reputation Analysis
2015-08 Jie Jiang (TUD)
Organizational Compliance: An agent-based model for designing and evaluating organizational interactions
2015-09 Randy Klaassen (UT)
HCI Perspectives on Behavior Change Support Systems
2015-10 Henry Hermans (OUN)
OpenU: design of an integrated system to support lifelong learning

2015-11 Yongming Luo (TUE)
Designing algorithms for big graph datasets: A study of computing bisimulation and joins

2015-12 Julie M. Birkholz (VU)
Modi Operandi of Social Network Dynamics: The Effect of Context on Scientific Collaboration Networks

2015-13 Giuseppe Procaccianti(VU)
Energy-Efficient Software

2015-14 Bart van Straalen (UT)
A cognitive approach to modeling bad news conversations

2015-15 Klaas Andries de Graaf (VU)
Ontology-based Software Architecture Documentation

2015-16 Changyun Wei (UT)
Cognitive Coordination for Cooperative Multi-Robot Teamwork

2015-17 André van Cleeff (UT)
Physical and Digital Security Mechanisms: Properties, Combinations and Trade-offs

2015-18 Holger Pirk (CWI)
Waste Not, Want Not! - Managing Relational Data in Asymmetric Memories

2015-19 Bernardo Tabuenca (OUN)
Ubiquitous Technology for Lifelong Learners

2015-20 Loïs Vanhée(UU)
Using Culture and Values to Support Flexible Coordination

2015-21 Sibren Fetter (OUN)
Using Peer-Support to Expand and Stabilize Online Learning

2015-22 Zhemin Zhu (UT)
Co-occurrence Rate Networks

2015-23 Luit Gazendam (VU)
Cataloguer Support in Cultural Heritage

2015-24 Richard Berendsen (UVA)
Finding People, Papers, and Posts: Vertical Search Algorithms and Evaluation

2015-25 Steven Woudenberg (UU)
Bayesian Tools for Early Disease Detection

2015-26 Alexander Hogenboom (EUR)
Sentiment Analysis of Text Guided by Semantics and Structure

2015-27 Sándor Héman (CWI)
Updating compressed colomn stores

2015-28 Janet Bagorogoza (TiU)
KNOWLEDGE MANAGEMENT AND HIGH PERFORMANCE; The Uganda Financial Institutions Model for HPO

2015-29 Hendrik Baier (UM)
Monte-Carlo Tree Search Enhancements for One-Player and Two-Player Domains

2015-30 Kiavash Bahreini(OU)
Real-time Multimodal Emotion Recognition in E-Learning

2015-31 Yakup Koç (TUD)
On the robustness of Power Grids

2015-32 Jerome Gard (UL)
Corporate Venture Management in SMEs

2015-33 Frederik Schadd (TUD)
Ontology Mapping with Auxiliary Resources

2015-34 Victor de Graaf (UT)
Gesocial Recommender Systems

2015-35 Jungxao Xu (TUD)
Affective Body Language of Humanoid Robots: Perception and Effects in Human Robot Interaction

====

2016

====

2016-01 Syed Saiden Abbas (RUN)
Recognition of Shapes by Humans and Machines

2016-02 Michiel Christiaan Meulendijk (UU)
Optimizing medication reviews through decision support: prescribing a better pill to swallow

2016-03 Maya Sappelli (RUN)
Knowledge Work in Context: User Centered Knowledge Worker Support

2016-04 Laurens Rietveld (VU)
Publishing and Consuming Linked Data

2016-05 Evgeny Sherkhonov (UVA)
Expanded Acyclic Queries: Containment and an Application in Explaining Missing Answers

2016-06 Michel Wilson (TUD)
Robust scheduling in an uncertain environment

2016-07 Jeroen de Man (VU)
Measuring and modeling negative emotions for virtual training

2016-08 Matje van de Camp (TiU)
A Link to the Past: Constructing Historical Social Networks from Unstructured Data

2016-09 Archana Nottamkandath (VU)
Trusting Crowdsourced Information on Cultural Artefacts

2016-10 George Karafotias (VUA)
Parameter Control for Evolutionary Algorithms

2016-11 Anne Schuth (UVA)
Search Engines that Learn from Their Users

2016-12 Max Knobbout (UU)
Logics for Modelling and Verifying Normative Multi-Agent Systems

2016-13 Nana Baah Gyan (VU)
The Web, Speech Technologies and Rural Development in West Africa - An ICT4D Approach

2016-14 Ravi Khadka (UU)
Revisiting Legacy Software System Modernization

2016-15 Steffen Michels (RUN)
Hybrid Probabilistic Logics - Theoretical Aspects, Algorithms and Experiments

2016-16 Guangliang Li (UVA)
Socially Intelligent Autonomous Agents that Learn from Human Reward

2016-17 Berend Weel (VU)
Towards Embodied Evolution of Robot Organisms

2016-18 Albert Meroño Peñuela (VU)
Refining Statistical Data on the Web

2016-19 Julia Efremova (TU/e)
Mining Social Structures from Genealogical Data

2016-20 Daan Odijk (UVA)
Context & Semantics in News & Web Search

2016-21 Alejandro Moreno Célleri (UT)
From Traditional to Interactive Playspaces: Automatic Analysis of Player Behavior in the Interactive Tag Playground

2016-22 Grace Lewis (VU)
Software Architecture Strategies for Cyber-Foraging Systems

2016-23 Fei Cai (UVA)
Query Auto Completion in Information Retrieval

2016-24 Brend Wanders (UT)
Repurposing and Probabilistic Integration of Data; An Iterative and data model independent approach

2016-25 Julia Kiseleva (TU/e)
Using Contextual Information to Understand Searching and Browsing Behavior

2016-26 Dilhan Thilakarathne (VU)

In or Out of Control: Exploring Computational Models to Study the Role of Human Awareness and Control in Behavioural Choices, with Applications in Aviation and Energy Management Domains

2016-27 Wen Li (TUD)
Understanding Geo-spatial Information on Social Media

2016-28 Mingxin Zhang (TUD)
Large-scale Agent-based Social Simulation - A study on epidemic prediction and control

2016-29 Nicolas Höning (TUD)
Peak reduction in decentralised electricity systems -Markets and prices for flexible planning

2016-30 Ruud Mattheij (UvT)
The Eyes Have It

2016-31 Mohammad Khelghati (UT)
Deep web content monitoring

2016-32 Eelco Vriezekolk (UT)
Assessing Telecommunication Service Availability Risks for Crisis Organisations

2016-33 Peter Bloem (UVA)
Single Sample Statistics, exercises in learning from just one example

2016-34 Dennis Schunselaar (TUE)
Configurable Process Trees: Elicitation, Analysis, and Enactment

2016-35 Zhaochun Ren (UVA)
Monitoring Social Media: Summarization, Classification and Recommendation

2016-36 Daphne Karreman (UT)
Beyond R2D2: The design of nonverbal interaction behavior optimized for robot-specific morphologies

2016-37 Giovanni Sileno (UvA)
Aligning Law and Action - a conceptual and computational inquiry

2016-38 Andrea Minuto (UT)
MATERIALS THAT MATTER - Smart Materials meet Art & Interaction Design

2016-39 Merijn Bruijnes (UT)
Believable Suspect Agents; Response and Interpersonal Style Selection for an Artificial Suspect

2016-40 Christian Detweiler (TUD)
Accounting for Values in Design

2016-41 Thomas King (TUD)
Governing Governance: A Formal Framework for Analysing Institutional Design and Enactment Governance

2016-42 Spyros Martzoukos (UVA)
Combinatorial and Compositional Aspects of Bilingual Aligned Corpora

2016-43 Saskia Koldijk (RUN)
Context-Aware Support for Stress Self-Management: From Theory to Practice

2016-44 Thibault Sellam (UVA)
Automatic Assistants for Database Exploration

2016-45 Bram van de Laar (UT)
Experiencing Brain-Computer Interface Control

2016-46 Jorge Gallego Perez (UT)
Robots to Make you Happy

2016-47 Christina Weber (UL)
Real-time foresight - Preparedness for dynamic innovation networks

2016-48 Tanja Buttler (TUD)
Collecting Lessons Learned

2016-49 Gleb Polevoy (TUD)
Participation and Interaction in Projects. A Game-Theoretic Analysis

2016-50 Yan Wang (UVT)
The Bridge of Dreams: Towards a Method for Operational Performance Alignment in IT-enabled Service Supply Chains

====
2017
====

2017-01 Jan-Jaap Oerlemans (UL)
Investigating Cybercrime
2017-02 Sjoerd Timmer (UU)
Designing and Understanding Forensic Bayesian Networks using Argumentation
2017-03 Daniël Harold Telgen (UU)
Grid Manufacturing; A Cyber-Physical Approach with Autonomous Products and Reconfigurable Manufacturing Machines
2017-04 Mrunal Gawade (CWI)
MULTI-CORE PARALLELISM IN A COLUMN-STORE
2017-05 Mahdieh Shadi (UVA)
Collaboration Behavior
2017-06 Damir Vandic (EUR)
Intelligent Information Systems for Web Product Search
2017-07 Roel Bertens (UU)
Insight in Information: from Abstract to Anomaly
2017-08 Rob Konijn (VU)
Detecting Interesting Differences:Data Mining in Health Insurance Data using Outlier Detection and Subgroup Discovery
2017-09 Dong Nguyen (UT)
Text as Social and Cultural Data: A Computational Perspective on Variation in Text
2017-10 Robby van Delden (UT)
(Steering) Interactive Play Behavior
2017-11 Florian Kunneman (RUN)
Modelling patterns of time and emotion in Twitter #anticipointment
2017-12 Sander Leemans (TUE)
Robust Process Mining with Guarantees
2017-13 Gijs Huisman (UT)
Social Touch Technology - Extending the reach of social touch through haptic technology
2017-14 Shoshannah Tekofsky (UvT)
You Are Who You Play You Are: Modelling Player Traits from Video Game Behavior
2017-15 Peter Berck, Radboud University (RUN)
Memory-Based Text Correction
2017-16 Aleksandr Chuklin (UVA)
Understanding and Modeling Users of Modern Search Engines
2017-17 Daniel Dimov (UL)
Crowdsourced Online Dispute Resolution
2017-18 Ridho Reinanda (UVA)
Entity Associations for Search
2017-19 Jeroen Vuurens (TUD)
Proximity of Terms, Texts and Semantic Vectors in Information Retrieval
2017-20 Mohammadbashir Sedighi (TUD)
Fostering Engagement in Knowledge Sharing: The Role of Perceived Benefits, Costs and Visibility
2017-21 Jeroen Linssen (UT)
Meta Matters in Interactive Storytelling and Serious Gaming (A Play on Worlds)
2017-22 Sara Magliacane (VU)
Logics for causal inference under uncertainty
2017-23 David Graus (UVA)
Entities of Interest--- Discovery in Digital Traces
2017-24 Chang Wang (TUD)
Use of Affordances for Efficient Robot Learning
2017-25 Veruska Zamborlini (VU)
Knowledge Representation for Clinical Guidelines, with applications to Multimorbidity Analysis and Literature Search

2017-26 Merel Jung (UT)
Socially intelligent robots that understand and respond to human touch
2017-27 Michiel Joosse (UT)
Investigating Positioning and Gaze Behaviors of Social Robots: People's Preferences, Perceptions and Behaviors
2017-28 John Klein (VU)
Architecture Practices for Complex Contexts
2017-29 Adel Alhuraibi (UVT)
From IT-BusinessStrategic Alignment to Performance: A Moderated Mediation Model of Social Innovation, and Enterprise Governance of IT
2017-30 Wilma Latuny (UVT)
The Power of Facial Expressions
2017-31 Ben Ruijl (UL)
Advances in computational methods for QFT calculations
2017-32 Thaer Samar (RUN)
Access to and Retrievability of Content in Web Archives
2017-33 Brigit van Loggem (OU)
Towards a Design Rationale for Software Documentation: A Model of Computer-Mediated Activity
2017-34 Maren Scheffel (OUN)
The Evaluation Framework for Learning Analytics
2017-35 Martine de Vos (VU)
Interpreting natural science spreadsheets
2017-36 Yuanhao Guo (UL)
Shape Analysis for Phenotype Characterisation from High-throughput Imaging
2017-37 Alejandro Montes Garcia (TUE)
WiBAF: A Within Browser Adaptation Framework that Enables Control over Privacy
2017-38 Alex Kayal (TUD)
Normative Social Applications
2017-39 Sara Ahmadi (RUN)
Exploiting properties of the human auditory system and compressive sensing methods to increase noise robustness in ASR
2017-40 Altaf Hussain Abro (VUA)
Steer your Mind: Computational Exploration of Human Control in Relation to Emotions, Desires and Social Support For applications in human-aware support systems"
2017-41 Adnan Manzoor (VUA)
Minding a Healthy Lifestyle: An Exploration of Mental Processes and a Smart Environment to Provide Support for a Healthy Lifestyle
2017-42 Elena Sokolova (RUN)
Causal discovery from mixed and missing data with applications on ADHD datasets
2017-43 Maaike de Boer (RUN)
Semantic Mapping in Video Retrieval
2017-44 Garm Lucassen (UU)
Understanding User Stories - Computational Linguistics in Agile Requirements Engineering
2017-45 Bas Testerink (UU)
Decentralized Runtime Norm Enforcement
2017-47 Jan Schneider (OU)
Sensor-based Learning Support
2017-46 Yie Yang (TUD)
Crowd Knowledge Creation Acceleration
2017-48 Angel Suarez (OU)
Collaborative inquiry-based learning

====

2018

====

2018-01 Han van der Aa (VUA)
Comparing and Aligning Process Representations
2018-02 Felix Mannhardt (TUE)
Multi-perspective Process Mining
2018-03 Steven Bosems (UT)
Causal Models For Well-Being: Knowledge Modeling, Model-Driven Development of Context-Aware Applications, and Behavior Prediction
2018-04 Jordan Janeiro (TUD)
Flexible Coordination Support for Diagnosis Teams in Data-Centric Engineering Tasks
2018-05 Hugo Huurdeman (UVA)
Supporting the Complex Dynamics of the Information Seeking Process
2018-06 Dan Ionita (UT)
Model-Driven Information Security Risk Assessment of Socio-Technical Systems
2018-07 Jieting Luo (UU)
A formal account of opportunism in multi-agent systems
2018-08 Rick Smetsers (RUN)
Advances in Model Learning for Software Systems
2018-09 Xu Xie (TUD)
Data Assimilation in Discrete Event Simulations
2018-10 Julienka Mollee (VUA)
Moving forward: supporting physical activity behavior change through intelligent technology
2018-11 Mahdi Sargolzaei (UVA)
Enabling Framework for Service-oriented Collaborative Networks
2018-12 Xixi Lu (TUE)
Using behavioral context in process mining
2018-13 Seyed Amin Tabatabaei (VUA)
Using behavioral context in process mining: Exploring the added value of computational models for increasing the use of renewable energy in the residential sector
2018-14 Bart Joosten (UVT)
Detecting Social Signals with Spatiotemporal Gabor Filters
2018-15 Naser Davarzani (UM)
Biomarker discovery in heart failure
2018-16 Jaebok Kim (UT)
Automatic recognition of engagement and emotion in a group of children
2018-17 Jianpeng Zhang (TUE)
On Graph Sample Clustering
2018-18 Henriette Nakad (UL)
De Notaris en Private Rechtspraak
2018-19 Minh Duc Pham (VUA)
Emergent relational schemas for RDF
2018-20 Manxia Liu (RUN)
Time and Bayesian Networks
2018-21 Aad Slootmaker (OUN)
EMERGO: a generic platform for authoring and playing scenario-based serious games
2018-22 Eric Fernandes de Mello Araujo (VUA)
Contagious: Modeling the Spread of Behaviours, Perceptions and Emotions in Social Networks
2018-23 Kim Schouten (EUR)
Semantics-driven Aspect-Based Sentiment Analysis
2018-24 Jered Vroon (UT)
Responsive Social Positioning Behaviour for Semi-Autonomous Telepresence Robots
2018-25 Riste Gligorov (VUA)
Serious Games in Audio-Visual Collections
2018-26 Roelof de Vries (UT)

Theory-Based And Tailor-Made: Motivational Messages for Behavior Change Technology

2018-27 Maikel Leemans (TUE)
Hierarchical Process Mining for Scalable Software Analysis

2018-28 Christian Willemse (UT)
Social Touch Technologies: How they feel and how they make you feel

2018-29 Yu Gu (UVT)
Emotion Recognition from Mandarin Speech

2018-30 Wouter Beek (VU)
The "K" in "semantic web" stands for "knowledge": scaling semantics to the web

====

2019

====

2019-01 Rob van Eijk (UL)
Web privacy measurement in real-time bidding systems. A graph-based approach to RTB system classification

2019-02 Emmanuelle Beauxis- Aussalet (CWI, UU)
Statistics and Visualizations for Assessing Class Size Uncertainty

2019-03 Eduardo Gonzalez Lopez de Murillas (TUE)
Process Mining on Databases: Extracting Event Data from Real Life Data Sources

2019-04 Ridho Rahmadi (RUN)
Finding stable causal structures from clinical data

2019-05 Sebastiaan van Zelst (TUE)
Process Mining with Streaming Data

2019-06 Chris Dijkshoorn (VU)
Nichesourcing for Improving Access to Linked Cultural Heritage Datasets

2019-07 Soude Fazeli (TUD)
Recommender Systems in Social Learning Platforms

2019-08 Frits de Nijs (TUD)
Resource-constrained Multi-agent Markov Decision Processes

2019-09 Fahimeh Alizadeh Moghaddam (UVA)
Self-adaptation for energy efficiency in software systems

2019-10 Qing Chuan Ye (EUR)
Multi-objective Optimization Methods for Allocation and Prediction

2019-11 Yue Zhao (TUD)
Learning Analytics Technology to Understand Learner Behavioral Engagement in MOOCs

2019-12 Jacqueline Heinerman (VU)
Better Together

2019-13 Guanliang Chen (TUD)
MOOC Analytics: Learner Modeling and Content Generation

2019-14 Daniel Davis (TUD)
Large-Scale Learning Analytics: Modeling Learner Behavior & Improving Learning Outcomes in Massive Open Online Courses

2019-15 Erwin Walraven (TUD)
Planning under Uncertainty in Constrained and Partially Observable Environments

2019-16 Guangming Li (TUE)
Process Mining based on Object-Centric Behavioral Constraint (OCBC) Models

2019-17 Ali Hurriyetoglu (RUN)
Extracting actionable information from microtexts

2019-18 Gerard Wagenaar (UU)
Artefacts in Agile Team Communication

2019-19 Vincent Koeman (TUD)
Tools for Developing Cognitive Agents

2019-20 Chide Groenouwe (UU)
Fostering technically augmented human collective intelligence

2019-21 Cong Liu (TUE)
Software Data Analytics: Architectural Model Discovery and Design Pattern Detection
2019-22 Martin van den Berg (VU)
Improving IT Decisions with Enterprise Architecture
2019-23 Qin Liu (TUD)
Intelligent Control Systems: Learning, Interpreting, Verification
2019-24 Anca Dumitrache (VU)
Truth in Disagreement- Crowdsourcing Labeled Data for Natural Language Processing
2019-25 Emiel van Miltenburg (UVT)
Pragmatic factors in (automatic) image description
2019-26 Prince Singh (UT)
An Integration Platform for Synchromodal Transport
2019-27 Alessandra Antonaci (OUN)
The Gamification Design Process applied to (Massive) Open Online Courses
2019-28 Esther Kuindersma (UL)
Cleared for take-off: Game-based learning to prepare airline pilots for critical situations
2019-29 Daniel Formolo (VU)
Using virtual agents for simulation and training of social skills in safety-critical circumstances
2019-30 Vahid Yazdanpanah (UT)
Multiagent Industrial Symbiosis Systems
2019-31 Milan Jelisavcic (VUA)
Alive and Kicking: Baby Steps in Robotics
2019-32 Chiara Sironi (UM)
Monte-Carlo Tree Search for Artificial General Intelligence in Games
2019-33 Anil Yaman (TUE)
Evolution of Biologically Inspired Learning in Artificial Neural Networks
2019-34 Negar Ahmadi (TUE)
EEG Microstate and Functional Brain Network Features for Classification of Epilepsy and PNES
2019-35 Lisa Facey-Shaw (OUN)
Gamification with digital badges in learning programming
2019-36 Kevin Ackermans (OUN)
Designing Video-Enhanced Rubrics to Master Complex Skills
2019-37 Jian Fang (TUD)
Database Acceleration on FPGAs
2019-38 Akos Kadar (OUN)
Learning visually grounded and multilingual representations

====
2020
====

2020-01 Armon Toubman (UL)
Calculated Moves: Generating Air Combat Behaviour
2020-02 Marcos de Paula Bueno (UL)
Unraveling Temporal Processes using Probabilistic Graphical Models
2020-03 Mostafa Deghani (UvA)
Learning with Imperfect Supervision for Language Understanding
2020-04 Maarten van Gompel (RUN)
Context as Linguistic Bridges
2020-05 Yulong Pei (TUE)
On local and global structure mining
2020-06 Preethu Rose Anish (UT)
Stimulation Architectural Thinking during Requirements Elicitation - An Approach and Tool Support
2020-07 Wim van der Vegt (OUN)
Towards a software architecture for reusable game components

2020-08 Ali Mirsoleimani (UL)
Structured Parallel Programming for Monte Carlo Tree Search
2020-09 Myriam Traub (UU)
Measuring Tool Bias & Improving Data Quality for Digital Humanities Research
2020-10 Alifah Syamsiyah (TUE)
In-database Preprocessing for Process Mining
2020-11 Sepideh Mesbah (TUD)
Semantic-Enhanced Training Data AugmentationMethods for Long-Tail Entity Recognition Models
2020-12 Ward van Breda (VU)
Predictive Modeling in E-Mental Health: Exploring Applicability in Personalised Depression Treatment
2020-13 Marco Virgolin (CWI)
Design and Application of Gene-pool Optimal Mixing Evolutionary Algorithms for Genetic Programming
2020-14 Mark Raasveldt (CWI/UL)
Integrating Analytics with Relational Databases
2020-15 Konstantinos Georgiadis (OU)
Smart CAT: Machine Learning for Configurable Assessments in Serious Games
2020-16 Ilona Wilmont (RUN)
Cognitive Aspects of Conceptual Modelling
2020-17 Daniele Di Mitri (OU)
The Multimodal Tutor: Adaptive Feedback from Multimodal Experiences
2020-18 Georgios Methenitis (TUD)
Agent Interactions & Mechanisms in Markets with Uncertainties: Electricity Markets in Renewable Energy Systems
2020-19 Guido van Capelleveen (UT)
Industrial Symbiosis Recommender Systems
2020-20 Albert Hankel (VU)
Embedding Green ICT Maturity in Organisations
2020-21 Karine da Silva Miras de Araujo (VU)
Where is the robot?: Life as it could be
2020-22 Maryam Masoud Khamis (RUN)
Understanding complex systems implementation through a modeling approach: the case of e-government in Zanzibar
2020-23 Rianne Conijn (UT)
The Keys to Writing: A writing analytics approach to studying writing processes using keystroke logging
2020-24 Lenin da Nobrega Medeiros (VUA/RUN)
How are you feeling, human? Towards emotionally supportive chatbots
2020-25 Xin Du (TUE)
The Uncertainty in Exceptional Model Mining
2020-26 Krzysztof Leszek Sadowski (UU)
GAMBIT: Genetic Algorithm for Model-Based mixed-Integer optimization
2020-27 Ekaterina Muravyeva (TUD)
Personal data and informed consent in an educational context
2020-28 Bibeg Limbu (TUD)
Multimodal interaction for deliberate practice: Training complex skills with augmented reality
2020-29 Ioan Gabriel Bucur (RUN)
Being Bayesian about Causal Inference
2020-30 Bob Zadok Blok (UL)
Creatief, Creatieve, Creatiefst
2020-31 Gongjin Lan (VU)
Learning better -- From Baby to Better
2020-32 Jason Rhuggenaath (TUE)
Revenue management in online markets: pricing and online advertising
2020-33 Rick Gilsing (TUE)
Supporting service-dominant business model evaluation in the context of business model innovation

2020-34 Anna Bon (MU)
Intervention or Collaboration? Redesigning Information and Communication Technologies for Development

2020-35 Siamak Farshidi (UU)
Multi-Criteria Decision-Making in Software Production

====
2021
====

2021-01 Francisco Xavier Dos Santos Fonseca (TUD)
Location-based Games for Social Interaction in Public Space

2021-02 Rijk Mercuur (TUD)
Simulating Human Routines: Integrating Social Practice Theory in Agent-Based Models

2021-03 Seyyed Hadi Hashemi (UVA)
Modeling Users Interacting with Smart Devices

2021-04 Ioana Jivet (OU)
The Dashboard That Loved Me: Designing adaptive learning analytics for self-regulated learning

2021-05 Davide Dell'Anna (UU)
Data-Driven Supervision of Autonomous Systems

2021-06 Daniel Davison (UT)
"Hey robot, what do you think?" How children learn with a social robot

2021-07 Armel Lefebvre (UU)
Research data management for open science

2021-08 Nardie Fanchamps (OU)
The Influence of Sense-Reason-Act Programming on Computational Thinking

2021-09 Cristina Zaga (UT)
The Design of Robothings. Non-Anthropomorphic and Non-Verbal Robots to Promote Children's Collaboration Through Play

2021-10 Quinten Meertens (UvA)
Misclassification Bias in Statistical Learning

2021-11 Anne van Rossum (UL)
Nonparametric Bayesian Methods in Robotic Vision

2021-12 Lei Pi (UL)
External Knowledge Absorption in Chinese SMEs

2021-13 Bob R. Schadenberg (UT)
Robots for Autistic Children: Understanding and Facilitating Predictability for Engagement in Learning

2021-14 Negin Samaeemofrad (UL)
Business Incubators: The Impact of Their Support

2021-15 Onat Ege Adali (TU/e)
Transformation of Value Propositions into Resource Re-Configurations through the Business Services Paradigm

2021-16 Esam A. H. Ghaleb (MU)
BIMODAL EMOTION RECOGNITION FROM AUDIO-VISUAL CUES

2021-17 Dario Dotti (UM)
Human Behavior Understanding from motion and bodily cues using deep neural networks

2021-18 Remi Wieten (UU)
Bridging the Gap Between Informal Sense-Making Tools and Formal Systems - Facilitating the Construction of Bayesian Networks and Argumentation Frameworks

2021-19 Roberto Verdecchia (VU)
Architectural Technical Debt: Identification and Management

2021-20 Masoud Mansoury (TU/e)
Understanding and Mitigating Multi-Sided Exposure Bias in Recommender Systems

2021-21 Pedro Thiago Timbó Holanda (CWI)
Progressive Indexes

2021-22 Sihang Qiu (TUD)

Conversational Crowdsourcing
2021-23 Hugo Manuel Proença (LIACS)
Robust rules for prediction and description
2021-24 Kaijie Zhu (TUE)
On Efficient Temporal Subgraph Query Processing
2021-25 Eoin Martino Grua (VUA)
The Future of E-Health is Mobile: Combining AI and Self-Adaptation to Create Adaptive E-Health Mobile Applications
2021-26 Benno Kruit (CWI & VU)
Reading the Grid: Extending Knowledge Bases from Human-readable Tables
2021-27 Jelte van Waterschoot (UT)
Personalized and Personal Conversations: Designing Agents Who Want to Connect With You
2021-28 Christoph Selig (UL)
Understanding the Heterogeneity of Corporate Entrepreneurship Programs

====

2022

====

2022-01 Judith van Stegeren (UT)
Flavor text generation for role-playing video games
2022-02 Paulo da Costa (TU/e)
Data-driven Prognostics and Logistics Optimisation: A Deep Learning Journey
2022-03 Ali el Hassouni (VUA)
A Model A Day Keeps The Doctor Away: Reinforcement Learning For Personalized Healthcare
2022-04 Ünal Aksu (UU)
A Cross-Organizational Process Mining Framework
2022-05 Shiwei Liu (TU/e)
Sparse Neural Network Training with In-Time Over-Parameterization
2022-06 Reza Refaei Afshar (TU/e)
Machine Learning for Ad Publishers in Real Time Bidding
2022-07 Sambit Praharaj (OU)
Measuring the Unmeasurable? Towards Automatic Co-located Collaboration Analytics
2022-08 Maikel L. van Eck (TU/e)
Process Mining for Smart Product Design
2022-09 Oana Andreea Inel (VUA)
Understanding Events: A Diversity-driven Human-Machine Approach
2022-10 Felipe Moraes Gomes (TUD)
Examining the Effectiveness of Collaborative Search Engines
2022-11 Mirjam de Haas (UT)
Staying engaged in child-robot interaction, a quantitative approach to studying preschoolers' engagement with robots and tasks during second-language tutoring
2022-12 Guanyi Chen (UU)
Computational Generation of Chinese Noun Phrases
2022-13 Xander Wilcke (VUA)
Machine Learning on Multimodal Knowledge Graphs: Opportunities, Challenges, and Methods for Learning on Real-World Heterogeneous and Spatially-Oriented
2022-14 Michiel Overeem (UU)
Evolution of Low-Code Platforms
2022-15 Jelmer Jan Koorn (UU)
Work in Process: Unearthing Meaning using Process Mining
2022-16 Pieter Gijsbers (TU/e)
Systems for AutoML Research
2022-17 Laura van der Lubbe (VUA)
Empowering vulnerable people with serious games and gamification
2022-18 Paris Mavromoustakos Blom (TiU)

Player Affect Modelling and Video Game Personalisation

2022-19 Bilge Yigit Ozkan (UU)
Cybersecurity Maturity Assessment and Standardisation

2022-20 Fakhra Jabeen (VUA)
Dark Side of the Digital Media - Computational Analysis of Negative Human Behaviors on Social Media

2022-21 Seethu Mariyam Christopher (UM)
Intelligent Toys for Physical and Cognitive Assessments

2022-22 Alexandra Sierra Rativa (TiU)
Virtual Character Design and its potential to foster Empathy, Immersion, and Collaboration Skills in Video Games and Virtual Reality Simulations

2022-23 Ilir Kola (TUD)
Enabling Social Situation Awareness in Support Agents

2022-24 Samaneh Heidari (UU)
Agents with Social Norms and Values - A framework for agent based social simulations with social norms and personal values

2022-25 Anna L.D. Latour (LU)
Optimal decision-making under constraints and uncertainty

2022-26 Anne Dirkson (LU)
Knowledge Discovery from Patient Forums: Gaining novel medical insights from patient experiences

2022-27 Christos Athanasiadis (UM)
Emotion-aware cross-modal domain adaptation in video sequences

2022-28 Onuralp Ulusoy (UU)
Privacy in Collaborative Systems

2022-29 Jan Kolkmeier (UT)
From Head Transform to Mind Transplant: Social Interactions in Mixed Reality

2022-30 Dean De Leo (CWI)
Analysis of Dynamic Graphs on Sparse Arrays

2022-31 Konstantinos Traganos (TU/e)
Tackling Complexity in Smart Manufacturing with Advanced Manufacturing Process Management