

Technical paper



## The SHOP4CF modular reference architecture for flexible process-oriented, data-driven smart manufacturing

Paul Grefen<sup>a,\*</sup>, Michał Zimniewicz<sup>b</sup>, Irene Vanderfeesten<sup>c</sup>, Kostas Traganos<sup>a</sup>,  
Pieter Becue<sup>d</sup>, Anders Pedersen<sup>e</sup>, Genesis Perez Rivera<sup>f</sup>

<sup>a</sup> Eindhoven University of Technology, Netherlands

<sup>b</sup> Poznan Supercomputing and Networking Center, Poland

<sup>c</sup> KU Leuven, Belgium

<sup>d</sup> IMEC, Leuven, Belgium

<sup>e</sup> Danish Technological Institute, Odense, Denmark

<sup>f</sup> FZI Research Center for Information Technology, Karlsruhe, Germany

### ARTICLE INFO

#### Keywords:

Smart manufacturing  
Reference architecture  
Manufacturing process management  
Manufacturing data management

### ABSTRACT

**Context:** Organizations in the smart industry domain face an increasing complexity of their functions and processes, both in the intra- and inter-organizational scopes. This has a direct effect on the digital systems that support their operations: they grow more complex too. At the same time, the organizations need to increase their agility: they must be able to flexibly adapt their processes to market changes. Especially SMEs in the manufacturing domain get lost in this combination of complexity and changeability.

**Objectives:** To help SME organizations in the smart manufacturing domain with their digital transformation, we develop the SHOP4CF modular reference architecture for digital manufacturing support.

**Methods:** We develop the SHOP4CF base architecture in an iterative way by application and evaluation in 36 real-world industrial cases, organized in three waves. We base our design partly on successful existing work, specifically the outcomes of the HORSE EU project, and align it with main manufacturing standards like ISA-95 and RAMI 4.0. We next distill the SHOP4CF reference architecture by abstracting the SHOP4CF base architecture, based on explicit design principles. We then specialize the reference architecture for process-oriented and data-driven manufacturing.

**Results:** The result of our work is a modular, flexible software reference architecture for smart manufacturing solutions. To facilitate its use, the reference architecture is coupled with manufacturing software life cycle models. Centered on a component marketplace, the life cycle for functional module developers is linked to the life cycle for module users, including explicit attention to the role of technology integrators. To illustrate its applicability, we describe three application cases in this paper.

**Conclusion:** The reference architecture provides a demonstrated point of departure for SMEs in the manufacturing domain to design their digital support in a complex and dynamic industry ecosystem. The modularity of the architecture and its coupling to software life cycles provide a new level of flexibility.

### 1. Introduction

Organizations operating in the smart industry domain are facing an increasing complexity of their functions and processes. This is the case in both the intra- and inter-organizational scopes. In the intra-organizational scope, shop floor processes become more complex as a consequence of growing complexity and required variability of manufactured products, leading from flexible manufacturing systems (FMSs)

to re-configurable manufacturing systems (RMSs) that have modularity as a design cornerstone [1]. In the inter-organizational scope, supply chain processes become more complex as a consequence of increasing numbers of suppliers and customers and more tightly connected to allow for just-in-time operation [2]. This growing complexity has a direct effect on the digital systems that support their operations, as well as the data structures that they manage: they grow more complex too. At the same time, manufacturing organizations need to increase their agility:

\* Correspondence to: P.O. Box 513, Eindhoven 5600 MB, Netherlands.

E-mail address: [p.w.p.j.grefen@tue.nl](mailto:p.w.p.j.grefen@tue.nl) (P. Grefen).

<https://doi.org/10.1016/j.jmsy.2026.01.010>

Received 5 November 2025; Received in revised form 4 January 2026; Accepted 15 January 2026

Available online 20 January 2026

0278-6125/© 2026 The Authors. Published by Elsevier Ltd on behalf of The Society of Manufacturing Engineers. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

they must be able to flexibly adapt their processes to market changes.

Especially SMEs in the manufacturing domain get lost in this combination of complexity and changeability. Often, they either get stuck in the digitization of their operations or try to find a one-size-fits-all technology, neither of which leads to the required solution. To help these organizations in the digital transformation of their manufacturing support, we have developed the SHOP4CF modular reference architecture for digital manufacturing support in SMEs, which we present in this paper. A reference architecture in smart manufacturing is the structural basis for practically developing digital applications [3,4].

The modularity of our reference architecture reflects the modular nature of re-configurable manufacturing systems as mentioned above. To facilitate its practical use, the reference architecture is coupled to two manufacturing software life cycle models. Centered on a component marketplace, the life cycle for functional module developers is linked to the life cycle for module users, including explicit attention for the role of technology integrators. We show how our work is partly based on successful existing work, specifically the outcomes of the HORSE EU project [5], and how it aligns with main standards like ISA-95 [6] and RAMI 4.0 [7].

In this paper, we also show specializations of the SHOP4CF reference architecture for two main aspects of smart manufacturing: process-oriented control and data-driven decision making. Smart manufacturing requires end-to-end integration of processes at the intra- [8,9] and inter-organizational levels [2,10]. Process management can contribute to efficient manufacturing and thereby to sustainability [11], addressing an Industry 5.0 aspect in smart manufacturing. The use of the large volumes of data available nowadays in smart manufacturing offers great opportunities towards improved decision making and thereby increased competitiveness [12]. To demonstrate the applicability of the reference architecture, we discuss the use of the reference architecture in three case studies: two retrospective studies completed in the European SHOP4CF project, and one prospective study related to new developments in human-centric manufacturing.

In the remainder of this introductory section, we first discuss the context of the development of the architecture to put things in proper perspective. Next, we discuss the purpose of the reference architecture in detail. We conclude this section by outlining the structure of this paper.

### 1.1. Context: SHOP4CF project

SHOP4CF is a research and innovation project in the European Horizon 2020 framework. The goal of the project is advancing the use of digital technologies in smart manufacturing while paying explicit attention to small and medium enterprises. The project is positioned in Industry 4.0 and Industry 5.0 developments, emphasizing the connected factory concept, from the process and data perspectives and from the intra- and interorganizational perspectives, and human aspects in manufacturing, from the perspectives of human well-being and interoperability between human and automated resources on the manufacturing shop floor.

Following the experience of the HORSE EU project [3,5], the SHOP4CF project is set up as an architecture-centric project. Consequently, digital architecture is the backbone for creating structure in the project in both its process and its products. In doing so, the architecture of the HORSE project [5] is extended in the SHOP4CF project. Firstly, SHOP4CF extends the architecture with data processing to support data-driven manufacturing. Secondly, where HORSE is scoped on the intra-organizational perspective [3] focusing on cyber-physical systems [13], SHOP4CF adds the inter-organizational perspective. Thirdly, where HORSE uses an architecture with predefined functionality, SHOP4CF proposes a flexible architecture the functionality of which can be defined by the choosing functional components from a marketplace. The human-centric perspective of SHOP4CF is reflected in the development of specific functional components in this context. The overall architecture design caters for positioning these components in the

SHOP4CF base architecture, which has been developed in an iterative way. The final version of this base architecture is abstracted into the reference architecture presented in this paper. The entire development process is explained in Section 3 of this paper.

The SHOP4CF base architecture is based on contemporary technical communication standards between components. In the project, FIWARE [14] is chosen as the basis for communication in the developed prototype system. The reference architecture presented in this paper is technology-agnostic, however. We present FIWARE as a possible technical middleware solution, but do not restrict the architecture design to this choice. To facilitate this, we make an explicit distinction between the definition of the conceptual architecture and the technical architecture.

### 1.2. Purpose of the reference architecture

The SHOP4CF reference architecture presented in this paper is based on domain knowledge on smart manufacturing, experience from the execution of the SHOP4CF project (and its predecessor HORSE project), plus general styles and patterns used in architecture design. The reference architecture aims at making the learnings from the project sustainable by open dissemination. The SHOP4CF reference architecture is meant to be used as a blueprint in concrete smart manufacturing developments by adding application context details from these concrete developments. This is illustrated in Fig. 1, adapted from the work of Bass et al. [15] and Grefen [16].

The SHOP4CF reference architecture is positioned as a classical facilitation architecture for multiple organizations [17]: its aim is not to strictly standardize, but to guide organizations in an industrial domain (smart manufacturing in our case) towards proper structure of digital technology. This paper is set up from a multi-aspect approach to architecture design [16]. It mainly centers on the software aspect architecture but includes discussions of other aspect architectures: the data aspect, process aspect, and platform aspect architectures.

The reference architecture provides a classification of smart manufacturing digital functionality into a well-defined structure, which is the basis for interoperability between components that offer specific functionality towards an integrated solution in an industrial context. The relation of the reference architecture to the development of these integrated smart manufacturing solutions on the one hand and the development of components to be used in such solutions on the other hand, leads to a framework with components as modular building blocks that caters for adaptable and extensible architectures and hence solutions – in modern times with fast-changing markets, it is necessary to build for change [18]. The components are organized in a catalog at a marketplace, which functions as the organizational interface between

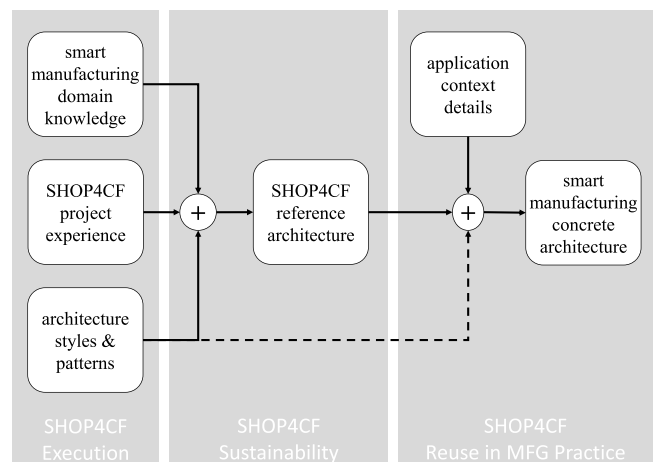


Fig. 1. global development and use of SHOP4CF reference architecture.

component builders and technology integrators – and in doing so, has the role of component quality insurance provider (we address this in the specification of the component life cycle in Section 4.2).

The reference architecture contains explicit positioning of functionality for three important and contemporary aspects of smart manufacturing solutions:

- support for process-centric manufacturing (end-to-end, integrated manufacturing process management);
- support for data-driven manufacturing (real-time, intelligent manufacturing to parameterize the execution of functions in processes);
- support for value-chain-level manufacturing management (factories of which the outputs and inputs are connected).

These points relate to the concept of *connected factory* in two complementary ways [19]. The former two points are related to the concept of *connected factory* in the intra-organizational sense, i.e., the *connected shop floor*. The latter point is related to the concept of *connected factory* in the inter-organizational sense, i.e., the *connected supply chain*.

The reference architecture presented in this paper is explicitly positioned with respect to industry standards to contribute to its general applicability. We cover the ISA-95 [6] and RAMI4.0 [7] standards in Section 2.1.

### 1.3. Structure of this paper

In Section 2 of this paper, we discuss the background of our work, the HORSE project preceding the SHOP4CF project, and general related work. In Section 3, we discuss the iterative development of the SHOP4CF base architecture that is used as input for the design of the SHOP4CF reference architecture. The SHOP4CF base architecture contains project-specific technology choices, which are abstracted from in the specification of the SHOP4CF reference architecture, as also explained in Section 3. In Section 4, we present the elaboration of life cycle management of software components in the context of the reference architecture. We distinguish between the life cycle of smart manufacturing solutions on the one hand and of smart manufacturing components on the other hand. As the reference architecture is based on components, we discuss the relation of the architecture to a component catalog in Section 4.3. In Section 5, we discuss the design and structure of the reference software architecture that we propose. We first pay attention to the starting points for the design and then continue presenting the logical reference architecture and technical reference architecture. In Section 6, we pay attention to the elaboration of the reference architecture towards two main aspects in smart manufacturing. Section 6.1 discusses process-driven smart industry support, i.e., shows how the reference architecture can be used to design end-to-end manufacturing process support. Section 6.2 discusses data-driven smart industry support, i.e., shows how the reference architecture is a basis in the design of smart manufacturing with decision making based on real-time data. We apply the reference architecture to a set of applications in Section 7. Here, we also discuss the lessons learnt in the design and use of the reference architecture. We conclude the paper in Section 8 by providing our main observations on the development of the reference architecture and presenting advice on the application of the architecture in practice.

## 2. Background

In this section, we present the background for the development of the reference architecture of the SHOP4CF project. In Section 2.1 we present standards that are relevant as a contextualization of our work. We specifically pay attention to the ISA-95 and RAMI 4.0 standards. In Section 2.2, we discuss related work. In Section 2.3, we briefly present the HORSE project, which is content-wise the predecessor of the SHOP4CF project and has inspired several important design choices in the project.

### 2.1. Background Standards

The ISA-95 standard [6] has been adopted by the International Society of Automation (ISA) as the standard for the interface content between manufacturing operations, operational control functions, and enterprise-level functions. It has been adopted by the ANSI and IEC organizations as well. The standard includes a manufacturing hierarchy that is widely used to position developments in the manufacturing domain. This hierarchy with five levels is shown in Fig. 2. It ranges from the smallest units of a manufacturing organization at the bottom of the hierarchy to the enterprise level at the top of the hierarchy.

The two lowest levels of the ISA-95 hierarchy are specialized towards four types of manufacturing activities (as indicated by the four columns in the figure):

- producing goods in batch production,
- producing goods in continuous production,
- producing goods in discrete production,
- storage and movement of goods.

Following the scope of the SHOP4CF project, the reference architecture presented in this paper focuses on the management of discrete manufacturing processes, where production lines are composed of work cells. Production lines are organized into production areas. One or more areas constitute a manufacturing site (i.e., one geographic location), one or more sites a manufacturing enterprise. In the context of ISA-95, the term *connected factory* is often interpreted as connecting individual work cells (and the equipment in these work cells) at the production line, area or site level. Or to phrase this differently: a connected factory is a factory in which all very local production facilities are digitally connected into an integrated shop floor.

The complexity of Industry 4.0 has called for an extension of the ISA-95 model that places its hierarchy in a broader perspective. This extension is realized in the Reference Architectural Model for Industry 4.0, or RAMI 4.0 in short [7]. RAMI 4.0 distinguishes between three dimensions to capture the complexity of Industry 4.0 developments. A graphical overview of this standard is shown in Fig. 3. We discuss it below.

As shown in Fig. 3, the three dimensions of RAMI 4.0 are labeled as *layers*, *life cycle & value stream* and *hierarchy levels*. These three dimensions have the following meaning:

- 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

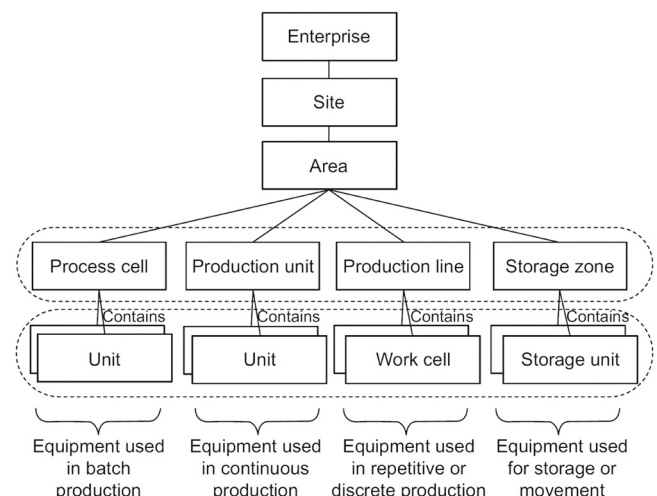


Fig. 2. ISA-95 reference hierarchy [6].

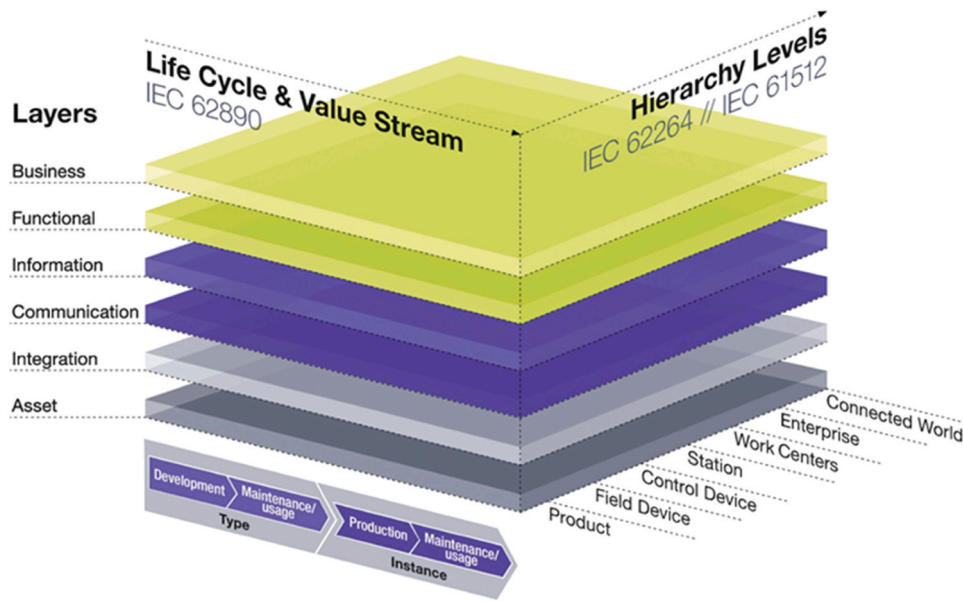


Fig. 3. RAMI 4.0 reference framework [7].

enterprise architecture. The dimension distinguishes the business, functional, information, communication, integration, and asset layers.

- 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 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. This dimension is related to the ISA-95 manufacturing hierarchy standard discussed above. The connected world level is introduced above the enterprise level of ISA-95 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.

Other than in an ISA-95 context, in a RAMI 4.0 context, the term *connected factory* is often used to describe a factory that is connected to other organizations, i.e., is digitally integrated at the *connected world* level of the *hierarchy levels* dimension. To avoid confusion between the two interpretations, in this paper we use the terms *connected shop floor* (ISA-95 interpretation of *connected factory*) and *connected value chain* (RAMI 4.0 interpretation of *connected factory*).

## 2.2. Embedding in and comparison to related work

In the previous subsections, we have contextualized our work in the SHOP4CF project towards the definition of a smart industry reference architecture with relevant standards and the predecessor project that ‘set the tone’ for an architecture-centric approach to smart industry support. In this section, we provide a broader embedding into other research and industrial development. We first pay attention to the relation of our work to reference architectures in general. Then, we look at related work into reference architectures for the smart manufacturing domain. Finally, we compare our approach to smart manufacturing to ‘practical’ architectures in industry.

### 2.2.1. Reference architectures in general

The concept of reference architecture is used in a number of domains [16]. In this paper, however, we use it for the domain of software architectures [15,20]. A general and widely used classification of software reference architectures is presented by Angelov et al. [17]. In this work, reference architectures are classified according to three dimensions: their context, their goal, and their design. The context dimension defines where the architecture is intended to be used, who has defined it, and when it was defined relative to the technology it relates to. The goal dimension describes why the reference architecture was defined: to standardize concrete architectures or to facilitate (i.e., to guide and inspire) the design of concrete architectures. The design dimension classifies what is described in a reference architecture, the level of detail of the description, the level of concreteness of the description, and the representation of the description. Based on combinations of the values in the dimensions, several archetypes of reference architectures are identified. The goal of the reference architecture presented in this paper is facilitation. In the context dimension, it is aimed at use in multiple organizations, it is defined by a non-profit organization (the consortium of the SHOP4CF project, with funding by the European Commission), and it is based on existing technology (i.e., it is a classical reference architecture in the classification). Consequently, our proposed architecture is a Type 3 reference architecture.

In several software (or information system) subdomains, reference architectures have been defined. An example subdomain that is of relevance for the current paper (as process-orientation is one of the foci of our work) is the workflow management or business process management subdomain. The earliest well-accepted reference architecture in this subdomain is that of the Workflow Management Coalition (WfMC) [21]. This reference architecture is rather coarse from a system point of view, however. The Mercurius reference architecture [22] is a more detailed reference architecture for workflow management systems containing several aggregation levels [16]. The S3 reference architecture [23] is aimed at the service orchestration field, defining a stratified set of component classes for business process management in a service-oriented architecture (SOA) setting. BPMS-RA [24] is a more recent definition of a reference architecture for business process management that defines sets of functional modules and their interfaces at several aggregation levels. The architecture proposed in this paper has specific similarities with all these existing architectures, which hence can be regarded as inspirational sources for our work.

### 2.2.2. Reference architectures for smart manufacturing

The Industrial Internet Reference Architecture (IIRA) [25] provides a functional viewpoint with five elements (or layers): business, application, information, operations, and control. These five elements can be mapped to the six elements in the Layers dimension of RAMI 4.0 [26]. The Chinese Intelligent Manufacturing System Architecture (IMSA) is a three-dimensional model [27] that has similarities to RAMI 4.0 but re-defines the Layers dimension as an Intelligent Functions dimension. IIRA and IMSA are, like ISA-95 and RAMI 4.0, more architecture frameworks than reference architectures, as they provide dimensions for positioning architecture elements, but do not suggest concrete system blueprints. As such, IIRA and IMSA are not directly comparable to the reference architecture presented in this paper, which aims to provide a structured system blueprint that directly guides the development of concrete smart manufacturing solutions. In this paper, we choose RAMI 4.0 as our guiding framework from the four discussed frameworks.

A review of several reference frameworks and architectures for smart manufacturing [26] identifies service-orientation as the main theme for development in smart manufacturing. In the analysis of this review, service-orientation fosters the offering of manufacturing resources as on-demand services at both the intra- and the inter-organizational level. The review [26] presents the evaluation of six propositions towards the adoption of service-orientation in smart manufacturing but does not propose a concrete reference architecture implementing this adoption. The choice for service-orientation points in the direction of flexible, component-oriented architectures [16], which is also the direction that we choose in the design of our reference architecture. We do distinguish, however, explicitly between the intra- and inter-organizational levels. For the intra-organizational level (i.e., the manufacturing shop floor level), we choose the composition of software components that are selected from a catalog to cater for an efficient and stable basis, rather than making basic shop floor manufacturing an ‘on-demand business’. For the inter-organizational level (i.e., the value network level), we choose dynamic service-oriented connections as well. Other than the review [26], we do present a guiding reference architecture to implement our direction of design.

The FIWARE smart industry reference architecture [14] presents a technical architecture that is technology-specific. It mainly provides a landscape for positioning various types of systems and technologies. The NIST service-oriented smart manufacturing system architecture [28] has a similar character, be it that it is not dedicated to a specific middleware type but is centered around a general manufacturing service bus concept. The IBM architecture for Industry 4.0 [29] provides a landscape for information technology components in smart manufacturing. It distinguishes between the equipment/device layer representing the edge side of smart manufacturing and the platform layer representing the hybrid cloud side of smart manufacturing. Its landscape character does not make it directly usable as a prescriptive reference architecture.

The HORSE reference architecture [3] can be seen as a predecessor of the reference architecture presented in this paper. It is less explicitly aimed at functional extensibility, however, lacks part of the functionality that is covered by the proposal in this paper (most specifically explicit support for data analysis), and is less explicitly positioned at the connected world level of the hierarchy dimension of the RAMI 4.0 framework. We discuss the HORSE project in more detail in Section 2.3. We position some of the discussed reference architectures and frameworks as explicit ‘predecessors’ in the development of the SHOP4CF reference architecture in Section 3.

### 2.2.3. Concrete architectures for smart manufacturing

In contemporary practice in large manufacturing organizations, digital support is often composed of a set of large application systems that fulfil specific functions in a manufacturing landscape. Where in the past custom-built, often monolithic systems were used to support manufacturing functions or simple processes, nowadays we see the use of common off-the-shelf (COTS) systems. Common examples of these

systems are product life cycle management (PLM) systems for the design and management of bills of materials (BoMs) and bills of processes (BoPs) [30], manufacturing execution systems (MES) for shop floor control [31,32], and production information management systems (PIMS) for management of data extracted from manufacturing activities.

Together, these systems are arranged and connected into a concrete architecture for manufacturing. The trend is to evolve these systems into smarter versions, such that they better support smart manufacturing scenarios. Commonly accepted reference architectures for the arrangement of these systems do not exist yet, however: there are vendor-specific architectures and there are informal system diagrams (designed either by individual manufacturing organizations or by integrator parties), but these are not fit for the solid design and engineering of vendor-neutral, flexible smart manufacturing architectures.

Also, many of the COTS systems mentioned above are too complex and costly for many SMEs in the manufacturing domain. For this reason, we design a reference architecture that has a solid structure, is vendor neutral, and allows the composition of smaller modules to cater to the SME world in smart manufacturing.

### 2.3. Predecessor project: HORSE

The HORSE project [3,5] is the most relevant predecessor project to the SHOP4CF project, in which the reference architecture has been developed that is presented in this paper. HORSE is a project in the Horizon 2020 program of the European Union, positioned in the Factories of the Future work program [33], focusing on ICT Innovation for Manufacturing SMEs. The goal of the HORSE project is to advance the industrial use of smart manufacturing by the development of an integrated framework that extends and unifies state-of-the-art technologies like smart robotics, the internet-of-things and business process management technology, and to develop software components to implement this framework in practice. The focus of the project is on both technology and human aspects, emphasizing worker safety on the manufacturing shop floor. Like SHOP4CF, the HORSE project focuses on the discrete manufacturing domain, i.e., on companies that manufacture individual products or batches of individual products.

The work in the HORSE project has been strongly related to the development of a guiding architecture. The HORSE architecture has a multi-aspect character, following the UT5 aspect framework [5,16] that defines five aspects or points of view on complex architectures, balancing separation of concerns between and integration of these aspects. One of the five aspects of this architecture is the software architecture. The software architecture has been developed using a multi-level approach using the principle of stepwise refinement. The Level 2 software architecture is shown in Fig. 4. More detailed levels of the software architecture (as well as the other four aspect architectures) are documented in the book reporting the HORSE results [5].

The architecture shown in Fig. 4 contains four functional subsystems (shown as rectangles) divided across two columns and two layers. The two columns represent the manufacturing process design phase and the manufacturing process execution phase – related to the design respectively manufacturing phases of a product in a factory. Consequently, these two phases are related to the *life cycle & value stream* dimension of the RAMI 4.0 framework (see Fig. 3): they correspond to the *Development* and *Production* phases. The two layers represent the aggregation levels for which design respectively at which execution takes place: the level of a production line or connected set of production lines (Global) and the level of an individual work cell within a production line (Local). These levels are related to the Hierarchy Levels dimension of RAMI 4.0 (and hence to the levels of ISA-95): they correspond to the *Work Centers* respectively *Station* levels. Consequently, we have the following four subsystems in the HORSE architecture: the *design global* subsystem contains support for the design of end-to-end manufacturing processes and the data sets supporting these; the *design local* subsystem contains support for the design of manufacturing tasks (as activities in the

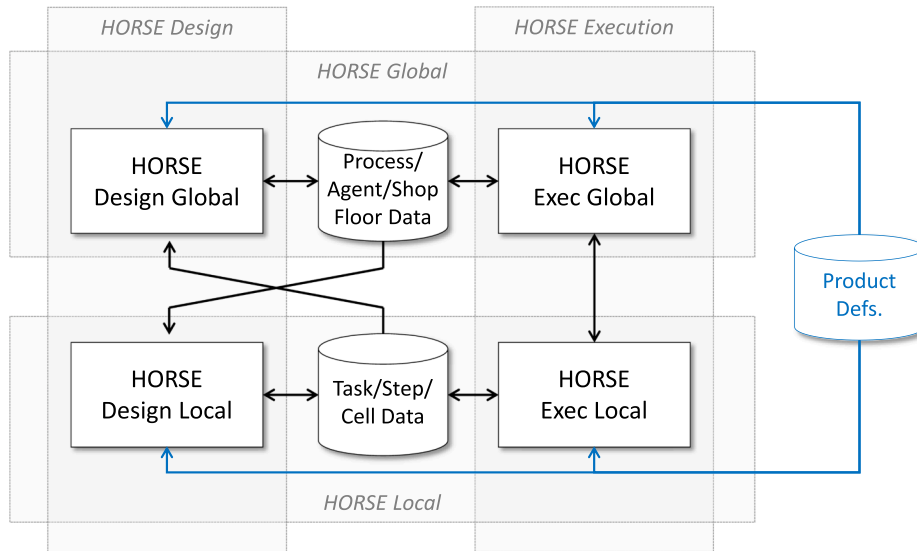


Fig. 4. HORSE Level 2 software architecture, adapted from [5].

processes) within manufacturing work cells and the data sets supporting these; the *execution global* subsystem provides support for the execution and monitoring of global processes; the *local execution* subsystem is aimed at supporting the execution of individual manufacturing tasks, including the coordination of multiple actors in a task (called a *team* in the HORSE approach [5]) and interfacing to human actors, robots, other machines, and sensors in work cells.

The four functional subsystems are connected to databases that contain definition and execution data at the Global and Local levels. They are also connected to a database that contains product definition data (typically from a product life cycle management (PLM) system [30] in a smart manufacturing environment). Development of this latter database is outside the scope of the HORSE project.

### 3. Development of the SHOP4CF reference architecture

The SHOP4CF reference architecture has been developed from the SHOP4CF base architecture, under the control of the SHOP4CF Architecture Team. In terms of the structure in Fig. 1, the SHOP4CF base architecture is represented by the left column in the figure as the input for the development of the reference architecture. In terms of its relation to the reference architectures and models discussed in Section 2, we can

position it as the newest incarnation of a lineage of models as shown in Fig. 5 – with other discussed models having some influence, but not an explicit relation. The WfMC reference architecture [21] provides a software structure blueprint for process management that has been elaborated in the Mercurius reference architecture [22], next applied in the HORSE reference architecture, and extended in the SHOP4CF reference architecture (the latter is explained in detail in Section 5.2 of this paper). ISA-95 [6] and RAMI 4.0 [7] are used to distinguish between the aggregation levels respectively the aspect dimensions in smart manufacturing. The FIWARE smart industry reference architecture [14] is used to bring the connectivity perspective, i.e., standardized middleware thinking, into the SHOP4CF reference architecture (without adopting the technical details of FIWARE, as the reference architecture is technology-agnostic). All these relations are summarized in the set of italicized arrow labels in Fig. 5.

The SHOP4CF base architecture was developed in an iterative fashion during the execution of the SHOP4CF project from 2020 to 2023. Fig. 6 shows an overview of the development process, in which the base architecture is labeled as ‘Arch.’ and the SHOP4CF Architecture Team as ‘SAT’. The development consisted of five phases, which resulted in the establishment of the SHOP4CF base architectures Version 0 to Version 4 [34]. The design of Version 0 was based on the SHOP4CF project charter and experience from previous projects (most notably the HORSE project as described in Section 2.3). Smart industry standards (most notably the ISA-95 and RAMI4.0 standards as described in Section 2.1 and indicated in Fig. 5) [19] and information system architecture patterns [16] were used in the design of this and consecutive versions. Version 1 was developed from Version 0 based on broad discussions in the SHOP4CF project consortium. Version 2 was developed by the application of Version 1 in four real-world industrial pilot cases within the bounds of the project consortium, and the evaluation of this application. Version 3 was developed by the application of Version 2 in the first wave of five consortium-external industrial application cases, which resulted from the first open call for application cases, and the evaluation of this application. Version 4 was developed by the application of Version 3 in the second wave of twenty-seven consortium-external industrial application cases, which resulted from the second and third open calls for application cases (for project-practical reasons these two calls were executed in parallel), and the evaluation of this application. Consequently, the iterative development of the SHOP4CF base architecture was steered by the evaluation based on 36 real-world industry cases across Europe, in which a large group of industry professionals was involved. The inclusion of two waves of open call cases catered for a

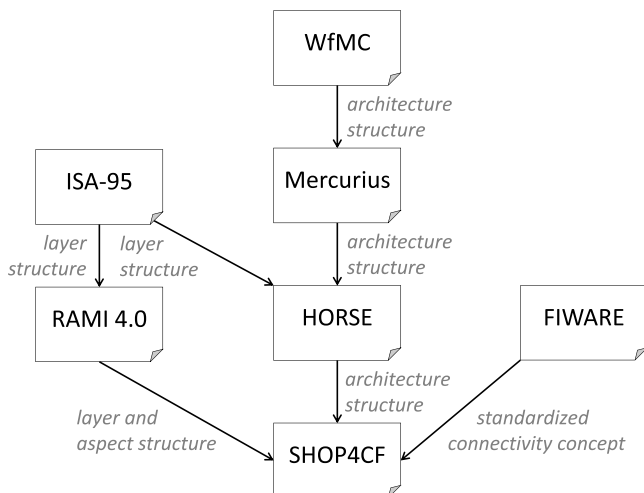


Fig. 5. SHOP4CF Reference Architecture in lineage of other reference models.

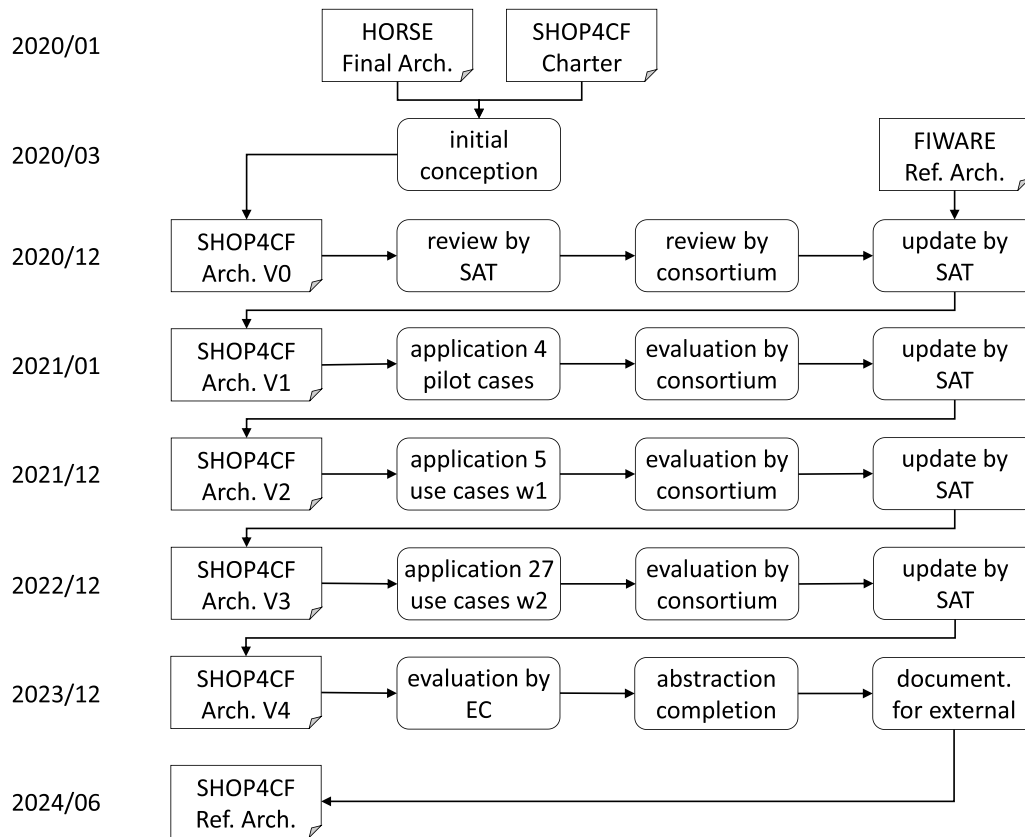


Fig. 6. iterative development of SHOP4CF base architecture and reference architecture.

broad spectrum of evaluative feedback on the architecture development, going beyond the ideas in the original project charter.

The final version of the SHOP4CF base architecture (Version 4) was evaluated by experts appointed by the European Commission. This version was next abstracted to remove project-specific details and completed (mostly with respect to explicit attention to the two artifact life cycles discussed in the next section of this paper) to form the SHOP4CF reference architecture for external dissemination. The main content changes to the various versions of the architecture (as described in Fig. 6) are summarized in Table 1; further details are provided in the SHOP4CF project deliverables [34,35].

The development of the pilot cases and the development of the open call cases were centered on realizing prototypes of smart manufacturing solutions. All open call cases were asked to report on the application of the architecture as part of the open call reporting process, which provided us in a structured way with feedback [34]. In these cases, however, the SHOP4CF architecture was a tool and not a goal. Therefore, we chose not to use a formal architecture evaluation approach like ATAM [36], because this would have complicated the collaboration with the use case owners whose main attention was on their solution. Instead, we have relied on an informal version of an action research approach, in which members of the SHOP4CF consortium collaborated closely with the use case owners to collect and record further feedback on the architecture and pass this to the SHOP4CF Architecture Team as requirements for the development of the next version of the architecture. In doing so, we targeted balancing scientific rigor and industrial relevance in our work as advocated by design science research in the information systems domain [37].

#### 4. Digital architecture and component life cycles

As discussed in the introduction, the SHOP4CF reference approach for agile smart manufacturing is based on the notion of a flexible system

Table 1

high-level overview of development of architecture across versions – linked to Fig. 6.

Version	Added/refined architecture elements	Rationale based on case input
V0	basic system architecture design; basic underpinning in Kruchten 4 + 1, UT5 frameworks; basic component lifecycle model; initial relation to ISA-95, RAMI 4.0, FIWARE frameworks	basic architecture setup – no case input yet
V1	elaboration relation to ISA-95, RAMI 4.0, FIWARE frameworks; mapping of initial prototype components set; mapping to pilot cases; initial data model considerations	better documentation for pilot case initialization; need for concrete software module design and application mappings; need for common data model basis
V2	update of initial prototype components set; better integration of FIWARE; first elaboration of data models	better alignment with pilot case needs in software setup; basis for practical data model design
V3	better alignment with IDS standard; alignment of data models with new standards; architecture use schemas for pilot cases	better hands-on guidance for development of w1 use cases; practical experience from w1 use cases
V4	alignment with deployment of pilot cases; update data models; updated alignment with industry standards (among others IDS) for increased interoperability	better hands-on guidance for development of w2 use cases; practical experience from w2 use cases
RA	abstraction from specific technology choices (e.g. FIWARE); split of lifecycle model into two perspectives; refinement of high-level data model	experiences from w1 and w2 use case deployments; practical feedback from EC experts; need to transform to reference architecture abstraction level

architecture that can evolve over time to reflect both changes in (external) market circumstances and learnings from (internal) manufacturing process execution analysis. This is enabled by the use of modular system components that can be flexibly integrated into an evolving architecture. The integration should be based on a proper specification of the syntax, semantics, behavior, and pragmatics of modules and a proper specification of the requirements following the same aspects from a potential user of a module.

The use of a modular architecture leads to the identification of two life cycles: the smart manufacturing system life cycle (where the manufacturer is in the lead) and the system component life cycle (where the component maker is in the lead). These two life cycles are interrelated. We discuss both below, as well as the relation between the two. The link between the two is created by a digital marketplace on which components are offered by component makers and bought by component integrators. The SHOP4CF project developed a specific marketplace (RAMP, see <https://ramp.eu>), but the life cycles discussed in this paper are not specialized towards any specific marketplace.

#### 4.1. Smart manufacturing system life cycle

The life cycle of a smart manufacturing system is shown in Fig. 7. We see that two roles are involved in this life cycle: the manufacturer (the organization that owns the system and specifies the requirements to the system) and the technology integrator (the organization that composes the system from components and integrates them into its technical context). The manufacturer has the primate role in this life cycle. Typical SME manufacturers have little in-house knowledge about realizing digital manufacturing systems and therefore rely on trusted, external technology integrators. Large manufacturers may have internal system development departments, in which case the technology integrator role is played by this department. In both cases, the technology integrator role hides details of the technology market from the manufacturer role: it is the organizational link to the system component life cycle discussed in Section 4.2. In principle, the manufacturer does not have to be aware of technical details of components.

The execution of an iteration of the life cycle of Fig. 7 is started by the *design* step of the manufacturer, which specifies the requirements to (a new evolution of) the smart manufacturing system – in practice often referred to as *solution design*. Note that this design step is different from that in Fig. 4: Fig. 7 is about designing a software system (a solution), Fig. 4 is about designing a use case for a software system (an application of the solution).

The design step is executed when new opportunities are identified in the external market (*identify*) or when structural evolution possibilities are identified in the analysis of the internal manufacturing process execution (*analyze*). The manufacturer then orders (the new version of) the smart manufacturing system with the technology integrator. The technology integrator creates a technical composition, which it may

iterate with the manufacturer. Then the designed composition is implemented in a system, integrated in its context, and tested. After successful testing by the technology integrator, the system is deployed at the manufacturer and goes through an acceptance test. After successful acceptance, the system goes into production execution. Parallel to this execution, data is produced and analyzed. The results of the analysis can be used for re-parameterization of the system in execution (on-the-fly evolution) or can be used for a structural upgrade of the system (re-design evolution).

#### 4.2. System component life cycle

Smart manufacturing systems (also referred to as smart manufacturing *solutions*) are created from system components. In the current timeframe, it is typically not desirable to develop ad-hoc components but to use pre-existing components: proven solutions are preferred for reasons of functionality, reliability and maintainability [38].

The life cycle of system components is shown in Fig. 8. The life cycle contains four roles: the component maker (which is the primate role in this life cycle), the marketplace on which components are traded, the technology integrator that integrates components into systems (to create smart manufacturing systems the life cycle of which is shown in Fig. 7) and the manufacturer that uses these systems. The two latter roles are the same roles as in Fig. 7. Note that we have chosen to include the manufacturer in the system component life cycle as this role owns the system of which components are integral parts. A technology integrator and manufacturer can post suggestions on the marketplace for new types or new versions of components (from a functional point of view) to further increase the standardization and reuse of functionality.

A component maker can make a new component (or adapt an existing component) based on one of three triggers: its own market analysis or observed technology evolution (*identify*), a suggestion from the marketplace (*suggest*), or a rejection of a component by the marketplace (*check*) because a component does not satisfy quality criteria. Here, the term ‘component’ is interpreted as ‘component type’ as it can be used multiple times across application systems. The *suggest* trigger is based on signals from technology integrators during the integration of components or suggestions from manufacturers that request new kinds of functionality. Both kinds of signals are channeled through the marketplace because of reasons of transparency and traceability. Components that are published by component makers are checked by the marketplace for conformance to the standards dictated by the marketplace – this provides a transparent level of quality assurance to component users. Components interpreted as ‘component instances’ are selected and retrieved by technology integrators. The *select* and *integrate* steps of the technology integrator role are part of the *compose* and *integrate* steps of Fig. 7: Fig. 7 has the per-system point of view (with usually multiple components), Fig. 8 has the per-component point of

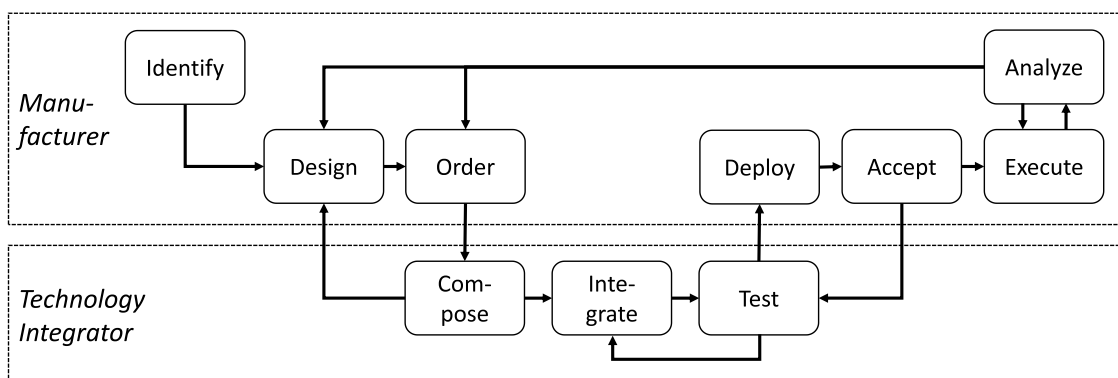


Fig. 7. smart manufacturing overall system life cycle (simplified).

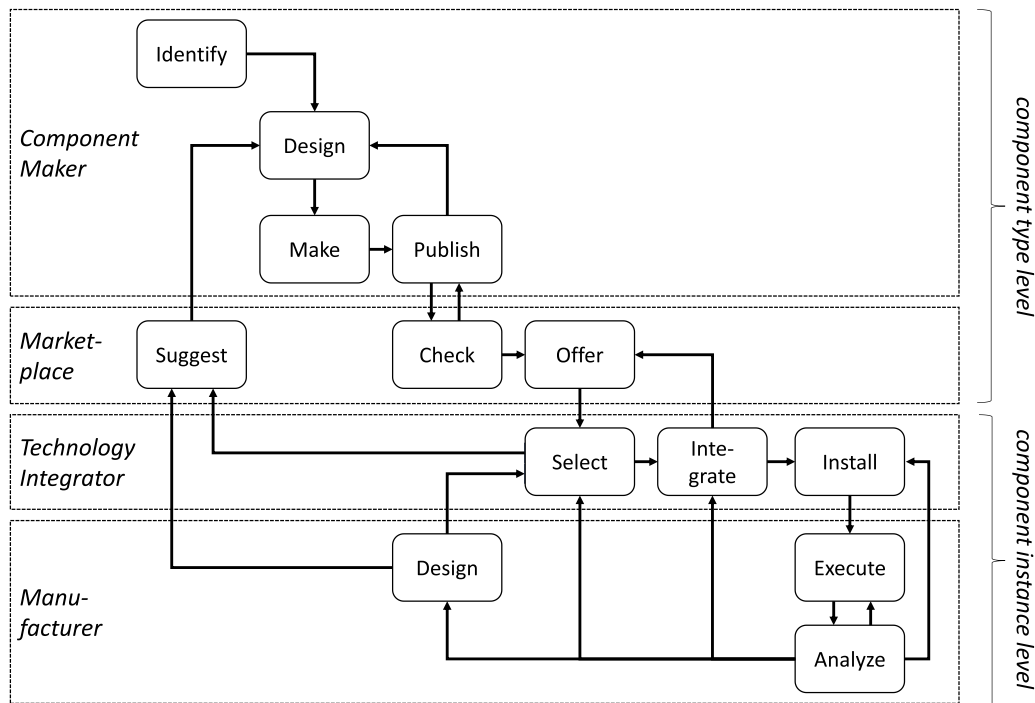


Fig. 8. smart manufacturing single system component life cycle (simplified).

view.

### 4.3. The marketplace

The marketplace is the organization that links the component life cycle and the system life cycle, as we have seen them in the previous two subsections. The marketplace manages the component catalog in which components are published and offered by component developers (as shown in Fig. 8), and from which components are selected and retrieved for integration by system integrators (as shown in Fig. 7).

The marketplace can be seen as a pivotal organization in a roadmap towards ‘plug-and-play’ composition of smart manufacturing solutions. As with all plug-and-play ecosystems, components have to be clearly defined and satisfy a set of constraints to be usable in the smart manufacturing solution market. These constraints operationalize the component concept.

From a technical point of view, components need to adhere to interface standards, process standards and data standards. Interface standards can be defined using an existing middleware standard to which components must comply to make them fit for integration into solutions. In the SHOP4CF project, the FIWARE [14] standard was chosen for this reason. Process standards define the way in which components communicate with other components from a domain functionality point of view. Data standards define the syntax and semantics of the contents of messages that components can exchange.

From a systems engineering and design point of view, components should be findable in a component catalog, i.e., they should be attributed with characteristics that describe what a system integrator is looking for – both in functional and non-functional terms. From the functional perspective, we have developed a partitioning of the functionality of smart manufacturing systems that functions as a high-level characteristic of components: each component should fit into one of the functional areas defined by the partitioning. The basis for this partitioning is the functional reference architecture that we present in the next section, which contains six main functional areas. The marketplace should check new components with respect to conformance with this partitioning (in the *check* activity of Fig. 8).

## 5. Overall reference architecture structure design

In this section, we discuss the design of the overall structure of the SHOP4CF reference architecture, based on the work towards the SHOP4CF base architecture as described in Section 3. In Section 5.1, we list the guiding principles that we have used for the design. We then focus first on the software aspect of the reference architecture (as the focal aspect of the UT5 framework mentioned in Section 2.3). In Section 5.2, we provide an overview of the structure of the logical reference architecture. In Section 5.3, we detail the architecture by adding interfaces and endpoints. In Section 5.4, we map the logical architecture to the technical reference architecture, describing the necessary technologies to embody it. In doing so, we make the connection to the *platform* aspect of the UT5 framework, which focuses on supporting technologies. In Section 5.5, we explain how the reference architecture is related to the connected world view of the RAMI 4.0 framework, i.e., we give it a broader scope. Then, to complement the software aspect of the reference architecture, we turn to the *data* aspect (which is a third aspect in the UT5 framework) of the reference architecture in Section 5.6.

### 5.1. Guiding principles for architecture design

The design of the SHOP4CF reference system architecture from the SHOP4CF base architecture follows a set of design guiding principles to obtain an overall design with a consistent basis and structure. Each of the guiding principles (GP) explicitly influences architecture design decisions in the elaboration of the architecture. The principles are based on general-purpose architecture principles from literature, e.g., from Greefhorst and Proper [38] and on main characteristics of smart manufacturing from a best practice point of view, documented for example by Grefen et al. [19]. These are the five guiding principles that we have formulated, together with their rationales:

1. *The architecture should support the definition and development of extensible digital manufacturing support by adhering to a strictly modular design.* This means that it should support the incremental inclusion of functional modules to enable evolving digital factories, while

safeguarding a clear overall solution structure. In doing so, it should comply with the life cycles presented in Section 4. Apart from the quality of extensibility, modularity is a basis for adequate reliability, maintainability and portability [38].

2. *The architecture should provide explicit support for data-driven smart manufacturing by including explicit data acquisition and processing functionality.* This means that it should support collecting shop floor data for use in manufacturing process improvement [39]. Data collection should preferably be executed in real-time to enable short feedback cycles.
3. *The architecture should provide explicit support for process-driven manufacturing enabling end-to-end process management in manufacturing.* The architecture should synchronize the activities of functional modules for manufacturing support in (near) real-time [19]. This follows the architecture principle of separating process logic from business logic [38].
4. *The architecture should provide explicit support for the connected factories concept at the supply chain level.* This means that the architecture should include structures to connect digital solutions of partners in a manufacturing ecosystem. In doing so, we make sure that the reference architecture covers the *Connected World* level in the *Hierarchy Levels* dimension of the RAMI 4.0 framework [7] (see Fig. 3). Note that the concept of *connected factory* at the shop floor level is directly supported by GP2 and GP3 and hence does not need coverage in GP4.
5. *The architecture should have a high degree of interoperability at both the internal and external levels.* This is to be achieved by adopting open standards to integrate with other approaches [16,38]. This should be facilitated by a choice of standard middleware and a standardized data architecture. Even though the SHOP4CF project has chosen a specific middleware standard (FIWARE [14]), the approach in this section is technology-agnostic.

### 5.2. Logical reference architecture design and overview

In this section, we design the logical reference architecture at the highest aggregation level: that of main functional subsystems. We use the Guiding Principles of Section 5.1 to guide our design, as discussed below.

To comply with GP1 formulated above, we use a modular, multi-level and multi-column architecture [16] like the one developed in the HORSE project [5] as discussed in Section 2.3 and shown in Fig. 4. We identify modular subsystems as containers for specific functionality, which can be chosen as concrete components depending on application context details (as indicated in Fig. 1).

To comply with GP2, we extend the main functional subsystems of the HORSE architecture (as shown in Fig. 4) by adding an Analyze column to the architecture that contains the Analyze Global and Analyze Local subsystems. These subsystems are logical containers for components dedicated to data processing, following the approach explained above. The Analyze Global subsystem is the container for functionality that processes data at the level that spans multiple work cells (like the shop floor level or enterprise level), the Analyze Local subsystem is the container for functionality that processes data of individual work cells.

To comply with GP3, we can rely on the existing structure of the HORSE architecture, which already covers process management functionality. Process support structures can have the form of functional modules within the identified subsystems. Following the design of the HORSE architecture [3,5,8], process design takes place at the global level for process definition and at the local level for task definition. Likewise, process execution takes place at the global level to coordinate task execution in an end-to-end fashion, at the local level for task execution. In the new Analyze column, process-oriented data analysis functionality can be added, like process mining functionality [40] at the global level and task mining functionality [41] at the local level.

To comply with GP4, we extend the HORSE architecture with an *External Ecosystem* layer. This layer contains the services of external

partners in each of three functional columns of the architecture. From a reference architecture point of view, these external services are seen as black boxes in the architecture (even though they can be embodied with mechanisms that implement gray boxes, like in the approach developed in the CrossWork project [2,42]). Note that the *Design Global* subsystem is not only triggered by partners in the ecosystem, but also by own observations in the market of the manufacturer (the *identify* activity in the life cycle of Fig. 7).

GP5 is not of consideration for the logical reference architecture and will hence be dealt with in the design of the technical reference architecture (see Section 5.4).

The above design considerations lead to the high-level logical reference architecture that is shown in Fig. 9.

The mapping of levels and columns in Fig. 9 to the ISA-95 and RAMI4.0 standards follows the same reasoning as that for the HORSE architecture in Section 2.3, except for two facts:

- The HORSE architecture does not consider the external ecosystem; as explained in GP4, the inclusion of the external ecosystem extends the coverage of the reference architecture with respect to the RAMI 4.0 framework.
- the Analyze column does not have an explicit mapping to the *life cycle & value stream* dimension of the RAMI 4.0 framework: within the context of a shop floor, it is part of the Production phase; in a connected world context, it is part of the Maintenance/Usage phase (we revisit this issue in Section 5.5).

The architecture subsystems shown in the two lower layers of Fig. 9 are embodied (i.e., mapped to concrete software modules) by the Design, Implement and Test activities of the Technology Integrator role in the life cycle diagram of Fig. 7. The concrete software components used in this embodiment are developed as shown in the component life cycle diagram of Fig. 8. This realizes GP1 listed in Section 5.1. Note that the subsystems in the external ecosystem layer of Fig. 9 are outside the scope of solution development and hence not embodied by the integrator as part of a manufacturing solution. Obviously, an integrator must cater for the interfaces to these subsystems.

### 5.3. Detailed logical reference architecture

Fig. 10 presents the detailed version of the logical reference architecture shown in Fig. 9. In this figure, we have detailed the interfaces between subsystems (following the architecture pattern class of independent components [15,16], added data repositories between subsystems (following the architecture pattern class of data-centered communication [15,16], and added the context of the manufacturing shop floor. In Fig. 10, the four left-most system modules and their two interconnecting databases (*SpecG* and *SpecL*) map architecture

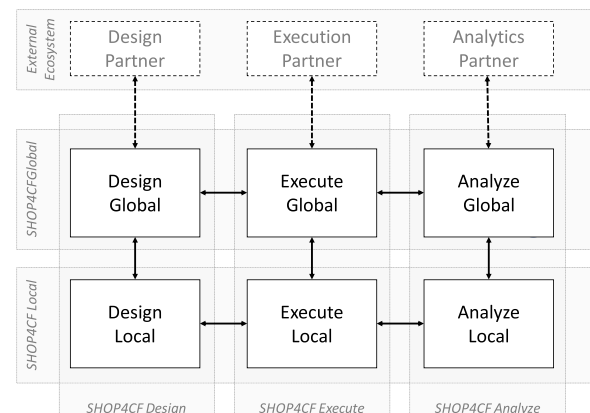


Fig. 9. high-level structure of the reference software architecture.

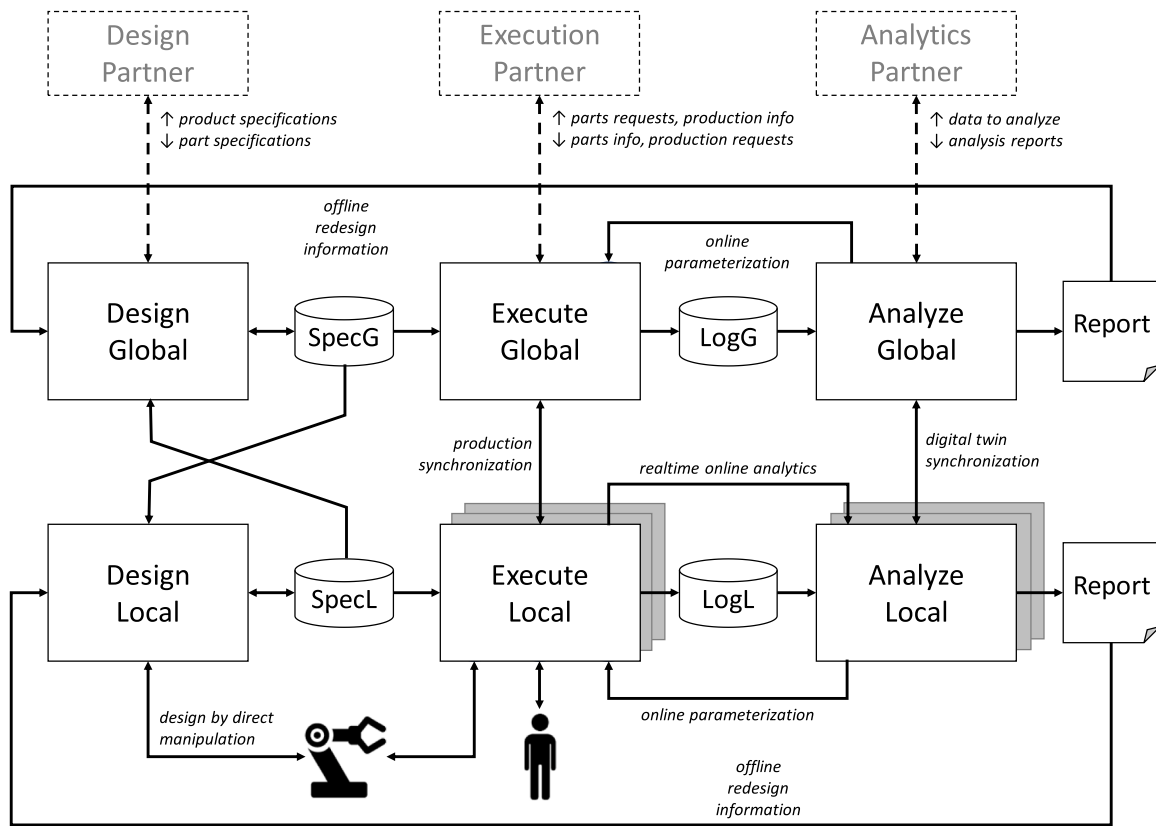


Fig. 10. subsystems in the reference logical system architecture with interfaces.

topology-wise in a one-to-one fashion to the four modules and the two databases of the HORSE Level 2 architecture shown in Fig. 4.

We make a note to Fig. 10. On the shop floor, we see human workers and automated (robotic) workers, indicated by the human and robot icons in the figure. Both types of workers can collaborate in teams (as specified in the HORSE concept model [5]). An example is a cobot, which is a team consisting of one human worker and one automated worker. We do not pay explicit attention to the aspect of teams, as this aspect is covered adequately in the HORSE project and is orthogonal to the issues discussed in this paper.

#### 5.4. Technical reference architecture

In the technical reference architecture, we outline the structure of the technical communication infrastructure between components that populate the six subsystems of the logical architecture. In doing so, we address GP5 as the last of the main design considerations presented in Section 5.1. We realize this by the introduction of middleware following common architecture principles [16,38].

At the level of the subsystems, i.e., that of the logical architecture shown in Fig. 9, we have the technical reference architecture shown in Fig. 11. In this architecture, we have introduced manufacturing middleware as the communication infrastructure between the six main subsystems of the reference architecture, replacing the point-to-point connections of the logical reference architecture. This middleware is tuned to support the real-time character of manufacturing processes – FIWARE [14] is an example of such middleware.

We have also introduced a layer between the subsystems at the global level and the external services. In this layer, we find enterprise middleware, enterprise applications and a firewall. Enterprise middleware is tuned towards more administrative applications – an enterprise service bus (ESB) is an example of such middleware. Enterprise applications include systems at Level 4 of the ISA-95 hierarchy (see Fig. 2),

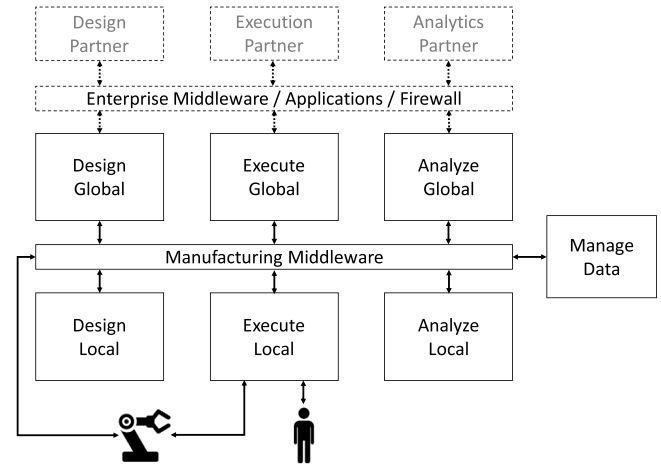


Fig. 11. reference technical system architecture (subsystem level).

like enterprise resource management (ERP) systems [43,44], product life cycle management (PLM) systems, customer relationship management (CRM) systems, and logistics management systems (LMS). In this paper, we abstract from the details of these systems as they are not relevant to the design of our shop floor-oriented reference architecture. Finally, we have introduced a technical subsystem for data management functionality required by the components in the six subsystems. This data management subsystem is connected to the manufacturing middleware as well for reasons of connection flexibility.

When we further detail the technical architecture of Fig. 11, we arrive at the technical reference architecture at the component level as shown in Fig. 12. Here, we see that the components that provide the functionality in the six functional subsystems are directly connected to

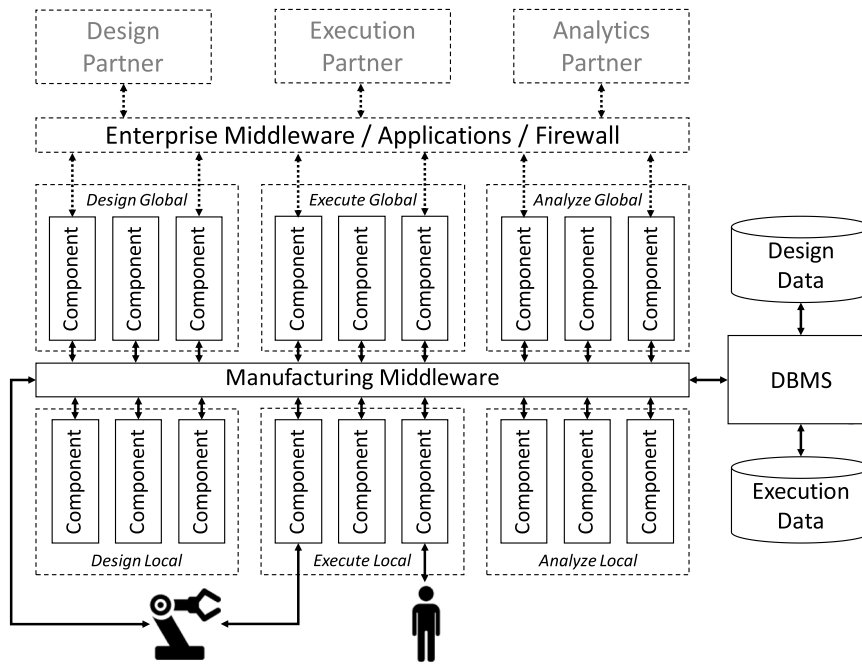


Fig. 12. reference technical system architecture (component level).

the manufacturing middleware to exchange data, forming a system of communicating processes [15,16]. In specific situations, there may be a direct connection between components (bypassing the middleware) if very strict, low-latency real-time communication is required – we omit this from the reference architecture for reasons of simplicity. Some components are also connected to the enterprise middleware to connect to application systems at Level 4 of the ISA-95 framework [6] or directly to systems of external partners. In principle, it is also possible to connect both types of middleware for this purpose by a middleware bridge, but for reasons of simplicity, we omit this discussion here. We see two databases in the figure, for design data and execution data, which map to the logical databases in Fig. 10.

The various sets of technical components in Fig. 12 can be linked to specific technology classes. For example, components in the *Execute Local* subsystem can be linked to the Industrial Internet of Things (IIoT) technology class [45] to enable flexible, nowadays often wireless, connections between physical resources on the manufacturing shop floor and other components in a solution.

5.5. The connected world view

After we have elaborated the intra-organizational part of the reference architecture, we now turn our attention to the elaboration of the inter-organizational part (explicitly addressing GP4 and GP5), i.e., of the External Ecosystem as identified in Fig. 9 and the Connected World level of the Hierarchy Levels dimension of the RAMI 4.0 framework as shown in Fig. 3. This inter-organizational level is becoming increasingly important, both because of the tighter integration of supply chains and because of the new business models that are made possible by the Industrial Internet of Things [46]. A tighter integration of supply chains allows for more just-in-time operations on the one hand and for increased resilience of manufacturing [47] at the chain level on the other hand. The modular nature of our reference architecture enables chain-level solutions to evolve as manufacturing markets evolve, thereby increasing flexibility and resilience levels. New business models can for example be based on networked products that provide their real-time status via digital channels – the automotive and aircraft industries provide well-known use cases here.

We operationalize the links to the three classes of partners of Fig. 9

by including technical links to these into the Connected World reference architecture. Design partners are connected by a Supply Chain Catalog Link, which provides access to catalogs of product design data. These catalogs can either be provided directly by supplier organizations or by digital platforms that represent an ecosystem of suppliers. Likewise, execution partners in the supply chain are connected by a Supply Chain Execution Link. Through this link, supply chain partners (either providers or customers) can be linked into chain-level business processes by exchanging supply chain process events – either directly or through a synchronization platform (for example a blockchain-based platform [48] that provides trust management in a supply chain). Analytics partners are connected by a Supply Chain Data Link. These can be partners that provide analytics services on submitted execution data, partners that provide data for internal analytics functions (including data from end users of manufactured products), and partners that require data to perform their analytics. To provide a general-purpose connection to International Data Spaces (IDS) platforms [49], an IDS connector is added and coupled directly to the manufacturing middleware, such that it can be accessed from any of the subsystems. The introduction of this connector follows the architecture principle that integration with external systems should be localized in a dedicated component [38].

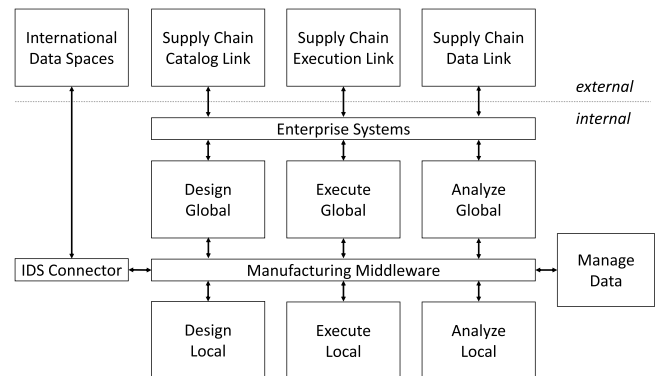


Fig. 13. connected world view of reference technical system architecture (subsystem level).

Fig. 13 shows that the internal-to-external link can be realized at two levels. The first level is at the manufacturing systems level, using the IDS connector and the manufacturing middleware. This level is used for external communication related to tasks that are part of the actual manufacturing process, i.e., are executed on the shop floor. The second level is at the enterprise level, either via internal enterprise-level systems (such as enterprise resource planning (ERP) systems) or via enterprise-level middleware (such as an enterprise service bus) – as detailed in Fig. 11. This level is used for tasks that are not executed on the shop floor, like procurement of parts or materials for manufacturing.

### 5.6. Data reference architecture

So far in this section, we have concentrated on the development of a software reference architecture for smart manufacturing. A complete reference architecture, however, consists of multiple aspects, of which the software aspect is one [16]. Several aspect frameworks exist that define sets of aspects or aspect dimensions of architectures. The UT5 framework [16] as used in the HORSE project [5], for example, is based on five dimensions: software, data, process, organization, and platform.

In the reference architecture presented in this paper, the data architecture is of primary importance next to the software architecture for two reasons. Firstly, because we have explicitly added data analysis functionality to our approach (as shown in Fig. 9), we need to address the exchange of data between design, execution and analysis system modules and hence need standardization following a reference format. Secondly, because we explicitly address the connected factory concept, we need to address the exchange of data between organizations in a manufacturing value chain and hence again need standardization.

Discussing a detailed data architecture goes beyond the level of detail of this paper, but an example is available in the SHOP4CF documentation [34]. Here, we discuss a high-level overview of a data reference architecture, as shown informally in Fig. 14. In the figure, lines denote content-based links and arrows denote *based-on* relations between classes. The structure consists of three layers that coincide with the functional pillars of the software reference architecture in Fig. 9. The *design data* level contains four groups of data classes that specify the work (processes) executed on a shop floor, the resources available to execute work (materials, workers and machines), the location of resources and work in progress, and events that are generated during the execution of work. All four groups are related as illustrated by the links in the top of the figure. For example, an event may be raised by a specific piece of work that is executed at a specific location by a specific

resource. The *execution data* contains instances of the design data, which are again interrelated (we show a few relations here to not clutter the figure). For example, a work instance describes the execution of a work definition for the creation of a single product. In practice, this is also referred to as a *Bill-of-Processes* (BoP) or *Bill-of-Manufacture* (BoMFR) [50]. Resource and location instance data may be related to physical characteristics of material resources and physical manufacturing locations (such as specific work cells) to enable direct linkage of the physical and digital worlds in high-speed, data-driven control mechanisms.

The groups of data classes shown in Fig. 14 can be further specialized (i.e., sub-typed). For reasons of brevity, we omit the full specializations in this paper. We do present an example of a specialization in Section 6.1, however.

The components of the *Design Global* and *Design Local* subsystems of Fig. 12 create (design), modify (redesign) and use (refer to) the contents (data items) of the *Design Data* layer of Fig. 14. The components of the *Execute Global* and *Execute Local* subsystems of Fig. 12 create (instantiate based on the definitions in the *Design Data* layer), modify (within the constraints defined in the *Design Data* layer) and use the contents (data items) of the *Execution Data* layer of Fig. 14. The components of the *Analyze Global* and *Analyze Local* subsystems of Fig. 12 transform contents (data items) in the *Execution Data* layer to produce higher-level data items for decision-making purposes that we label as information. These can either be within one of the four columns in the reference data architecture or can aggregate across columns. The latter we have labelled as *compound information*.

## 6. The process and data perspectives of smart manufacturing support

The SHOP4CF project was designed to pay explicit attention to two key aspects of smart manufacturing: support for end-to-end manufacturing processes [5,8] and data-driven manufacturing management [12,51]. Hence, we pay explicit attention to these two aspects as well in this paper: support for process management in Section 6.1 and support for data-driven manufacturing management in Section 6.2.

The importance and complementary character of these two perspectives is underlined by many common approaches in the information systems analysis research domain where the link between and integration of these two perspectives are extensively discussed and elaborated. In the field of enterprise modeling and enterprise architecture these perspectives are recognized as the two main lenses to look at an enterprise. For instance, they define the first two columns of the Zachman

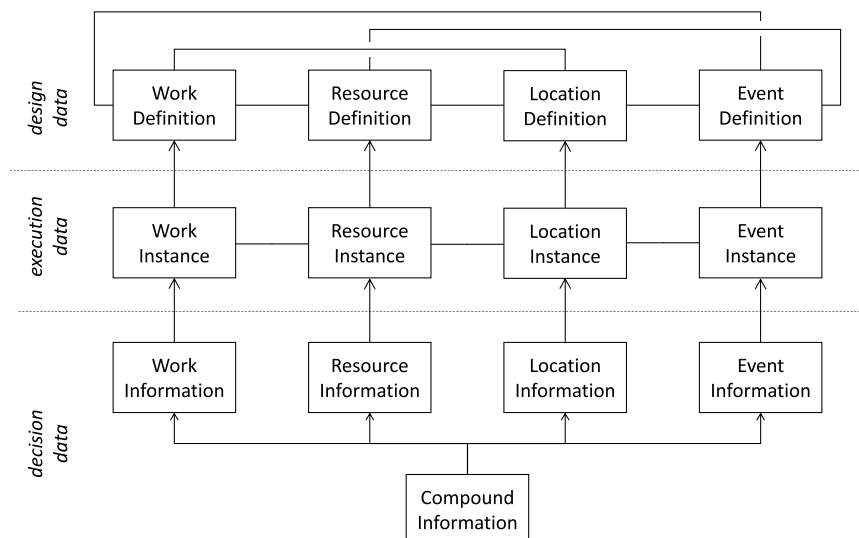


Fig. 14. high-level reference data architecture.

framework [52,53] describing the ‘what’ and the ‘how’ of the enterprise. The 4EM method [54] distinguishes a Concept (data) and a Business Process model as important components of an enterprise model, next to four other perspectives (Goals, Business Rules, Actors and Resources, Technical Components). Depending on the business challenges at hand for which the enterprise modeling is applied, one of these perspectives is leading in the analysis and creation of a composite enterprise model. In the field of business process management, substantial attention has been paid to the duality and integration of information (data) and activities (process), see for example the works by Kumaran et al. [55], Kumar [56], Reijers et al. [57], and Sun et al. [58].

6.1. Process-driven smart industry support

In this section, we show how end-to-end manufacturing processes are supported by a selection of modules in the reference architecture that we have presented in the previous section. We do this on the basis of the technical reference architecture as presented in Section 5.4 to stay close to the principle of marketplace-based, dynamic embodiment of the architecture as discussed on Section 4.3. The resulting specialization of the reference architecture for manufacturing process support is shown in Fig. 15.

In the architecture, the manufacturing middleware provides extensibility to process management functionality. As such, it can be seen as a more general implementation of the *software bus* structure proposed in the Mercurius reference architecture for workflow management [22].

We show an elaboration of the *Design Data* specific to manufacturing process management in Fig. 16 (based on the process concept model of the HORSE project [5] and relate it to the design modules in the software architecture of Fig. 15).

The *Process Designer* component creates instances of the *Manufacturing Process Definition* class. These are instantiated for interpretation by the *Process Executor* component, which provides process execution support for manufacturing processes [59]. The *Task Designer* creates instances of the *Manufacturing Tasks Definition* class. These are instantiated for interpretation by the three components in the *Execute Local* subsystem. These three components are responsible for instructing robotic actors, human actors, and the synchronization of teams that consist of one or more actors, either of the robotic or human class.

6.2. Data-driven smart-industry support

In this section, we show how data-driven manufacturing

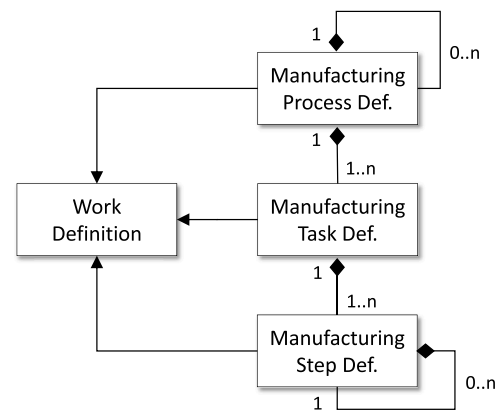


Fig. 16. specialization of class Work Definition (see Fig. 14).

management is supported by a selection of modules in the reference architecture presented in the previous section. Like for process support, we do this on the basis of the technical reference architecture as presented in Section 5.4 to stay close to the principle of marketplace-based, dynamic embodiment of the architecture as discussed in Section 4.3. The specialization of the reference architecture for data-driven manufacturing is shown in Fig. 17.

The components in the *design* column of the reference architecture on the one hand assist in making product designs respectively production step designs that enable the production of manufacturing data. On the other hand, these components use data distilled from the execution of manufacturing steps and processes in optimizing the design of products and manufacturing steps. In the *execute* column, we find the data collector modules on the global and local level of manufacturing. Data collection from robotic task execution proceeds typically quite differently from data collection from human activities. Hence, we have placed two data collector components at the local level. Data collection from robotic actors typically relies on digital interfaces that are integrated into the robot controllers. Data collection from human actors relies on several monitoring technologies, such as wearable IoT sensors [60], smart camera technology, and input that is explicitly provided by human operators (via interactive control interfaces). In the *analyze* column, we find the components that preprocess data (by filtering, normalization, and aggregation), analyze data (by performing statistical analyses and by recognizing patterns by automated learning), and present data by means of reports and dashboards. These reports and dashboards can be

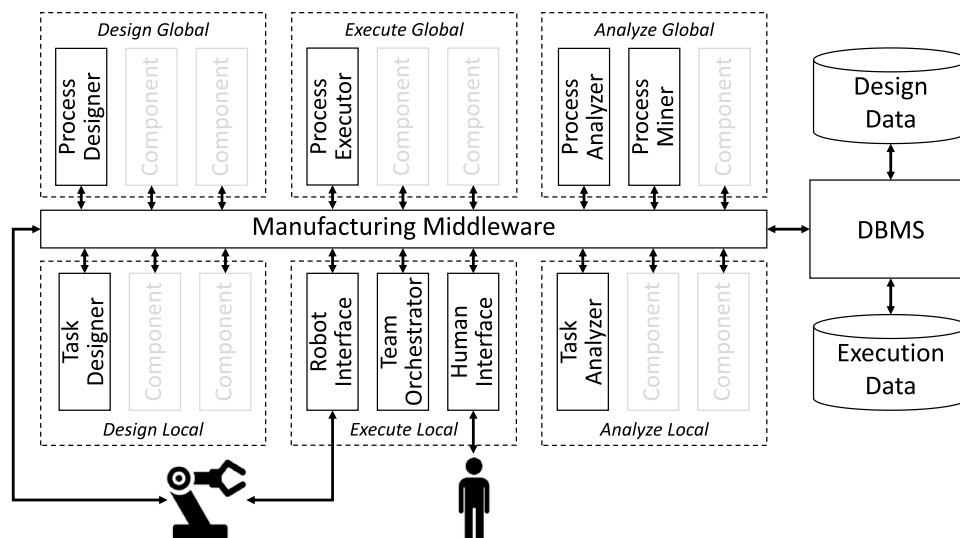


Fig. 15. specialized reference system architecture with process management components highlighted.

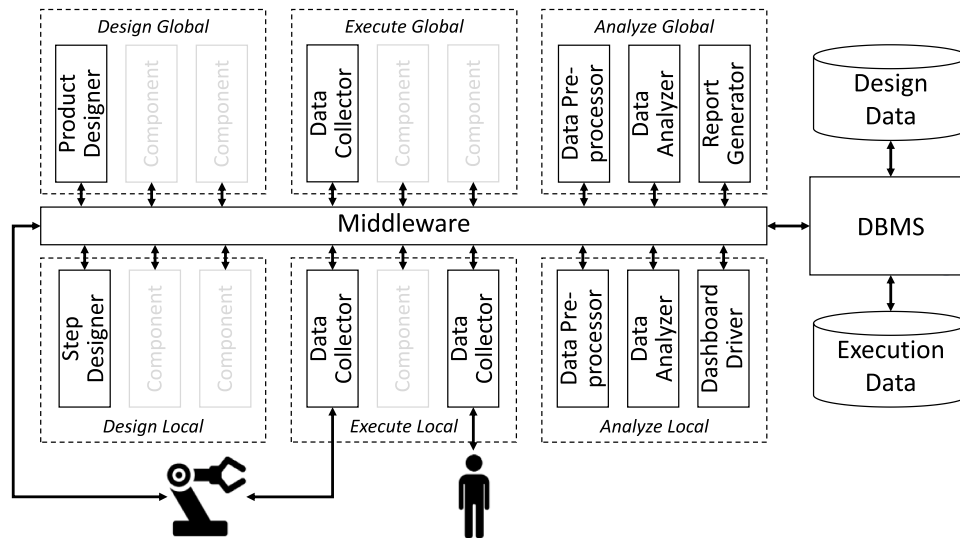


Fig. 17. specialized reference system architecture with data management components highlighted.

used for optimization of operations management, but also for the improvement of worker conditions in light of Industry 5.0 developments (the OOFA case that we discuss in Section 7.3 is an example of the latter).

Recent advances in the artificial intelligence (AI) field provide technology that can be used as a basis for these modules [61]. For example, automated time series summarization algorithms [62] can be used in the two data collector components to perform automated data aggregation functions. Diffusion models as a component of generative AI (GenAI) can be a basis for both data analyzer components, for example in fault diagnosis [63]. They can also be used in the dashboard driver component and the report generator component as part of intelligent controllers for operations management in adaptive manufacturing [63].

The interpretation of generated information can be performed by humans (like product and process designers) in traditional manufacturing settings. In highly automated, data-driven settings, the output of the data analyzer components may be directly coupled to the components in the design column, such that automated learning and evolution takes place in a feedback loop across the three columns. To enable the latter, standardized data models (based on the data reference architecture as discussed in Section 5.6) are essential ingredients to facilitate interoperability in such a feedback loop: they are the key to breaking the data silos found in legacy components. If the feedback loop is extended to the connected world, for example to enable customer outcome management in smart manufacturing [64], inter-organizational data flows need to be standardized (for example using GS1 reference models [65]).

### 7. Applying the reference architecture in smart industry projects

This section shows the application of the SHOP4CF reference architecture in structuring concrete smart industry applications (i.e., apply it as illustrated in Fig. 1). The architectures of such applications are typically ‘invented’ from scratch, i.e., without a solid basis in a standardized reference architecture. Sometimes they are ‘inspired’ by standards like ISA-95 [6] or RAMI 4.0 [7], but this is not an adequate basis for standardization to achieve modularity, reusability and interoperability of systems built from these architectures. In this section, we show how the reference architecture described in this paper is a basis for such standardization.

As described in Section 3, during the SHOP4CF project, 37 industrial use cases were developed [34,35]. Each use case was supported at least by a component developer and/or integrator plus an industrial partner,

where the components were to be deployed and evaluated in a realistic manufacturing setting. Using the SHOP4CF base architecture, most of the experiments integrated already existing SHOP4CF components from the marketplace catalog (i.e., following the life cycle shown in Fig. 7), together with newly developed components, which were either a modular extension of pre-existing components or designed and developed from scratch.

In this section, we take a retrospective look at a few use cases from the SHOP4CF reference architecture point of view as the final consolidated version of the base architecture used in the use cases. In Sections 7.1 and 7.2, we discuss two use cases from the SHOP4CF project (a simpler one and a more elaborate one) in retrospect to show how the reference architecture was used to refine (i.e., to standardize and complete) the initial architecture design of the solutions developed in these experiments.

To demonstrate the usability of the reference architecture beyond the scope of the SHOP4CF project, we also have applied the architecture to one prospective case that uses it as a blueprint to design a targeted smart industry application. We describe this in Section 7.3. We end this section with a short discussion of the overall learnings from applying the reference architecture to the set of use cases.

#### 7.1. SHOP4CF Case 1: RASP

In the open call series of the SHOP4CF project, RASP was an application project targeted at applying robot-based automation to a screen-printing process. The developed solution consists of two parts. The first part, supporting the execution (printing) phase of the overall process, consists of a robot for automatically handling light and thin sheets. The robot must be able to effectively grip these sheets and position them precisely for printing. The second part, supporting the post-execution (quality analysis) phase of the overall process, consists of an automated visual inspection mechanism that detects printing errors. The initial design of the RASP architecture is shown in Fig. 18.

From an architecting point of view, the initial RASP architecture shows two flaws. Firstly, it does not comply with architecture modularity rules: one component crosses sub-system boundaries and hence does not comply with a proper system-subsystem structure. Hence, it violates guiding principle GP1 for architecture design as discussed in Section 5.1. Secondly, the design column of the architecture is completely empty: this indicates that application design may be completely hardcoded in software design, which impedes easy evolution of functionality. This violates architecture design principle GP3.

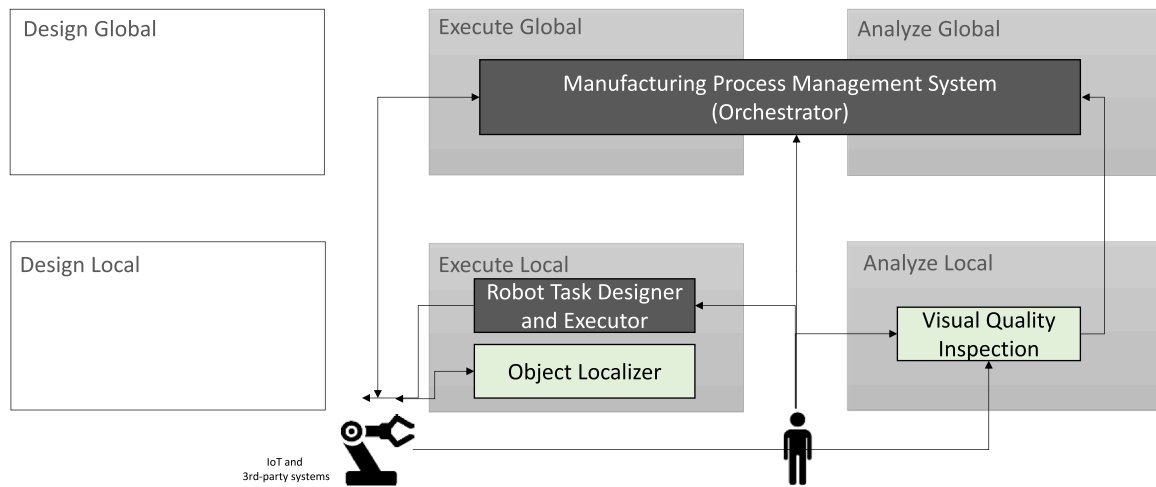


Fig. 18. RASP initial architecture design.

The reference architecture described in Section 5 was used as a tool to redesign the initial architecture and comply with all guiding principles. This led to several architecture redesign decisions. Firstly, the manufacturing process management system was split into two components. Secondly, the robot task designer and executor module was split into two modules to distinguish between the task design and task execution phases. Thirdly, the object localizer module was split into two modules that serve the Design Local and Analyze Local sub-systems. The resulting final design of the RASP architecture is shown in Fig. 19. In this figure, all used modules are shown, the four standard modules retrieved from the marketplace (as discussed in Section 2.3) are shown with a grayed fill, and the three newly developed modules are shown with a white fill.

The final architecture shows that the Design Global subsystem is left empty. The reason for this is that the process model for the process execution is ‘hardwired’ into the Process Executor module. This hardwiring of manufacturing processes is in general an approach that will cause flexibility problems in further development of the solution: process flow and task functionality need to be separated to realize flexibility towards evolution of a solution [8,9,59,66].

The benefits of the use of the SHOP4CF reference architecture in this case are threefold. Firstly, the use of existing system components in the smart manufacturing system life cycle as discussed in Section 4.1 leads to more than 50 % reduction in the development of new modules. By

adapting component granularity to the reference architecture, possible mappings of requirements to existing components become identifiable. Secondly, the more modular character of the redesigned architecture leads to more flexibility and interoperability of future modifications and extensions of the solution. Thirdly, the fact that one subsystem has remained empty in the redesigned architecture is a clear indication to the case owner where an important improvement of the solution is possible. Hence, the reference architecture helps in strategic development of the smart manufacturing solution towards better use of proven process management principles [8,9,59,66].

7.2. SHOP4CF Case 2: MATTRESS/HOPE\_Foreman/SISTERS

In the context of the open call experiments in the SHOP4CF project, a series of three application projects was executed throughout the three rounds of the open calls, called respectively MATTRESS, HOPE\_Foreman, and SISTERS. These application projects build on top of each other, extending the developed smart manufacturing systems through the iterations. The first two projects are targeted towards smart manufacturing of products made from bent pipes; the latter project is targeted towards smart manufacturing of custom plastic products.

The initial architecture resulting from the three iterations is shown in Fig. 20, with for each component an indication in which iteration it was added (and with color coding per iteration). The main component is

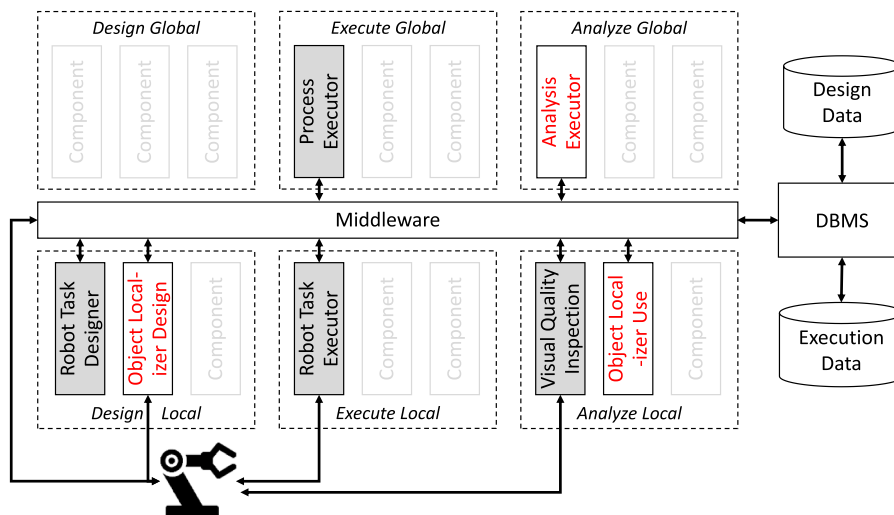


Fig. 19. RASP final architecture design.

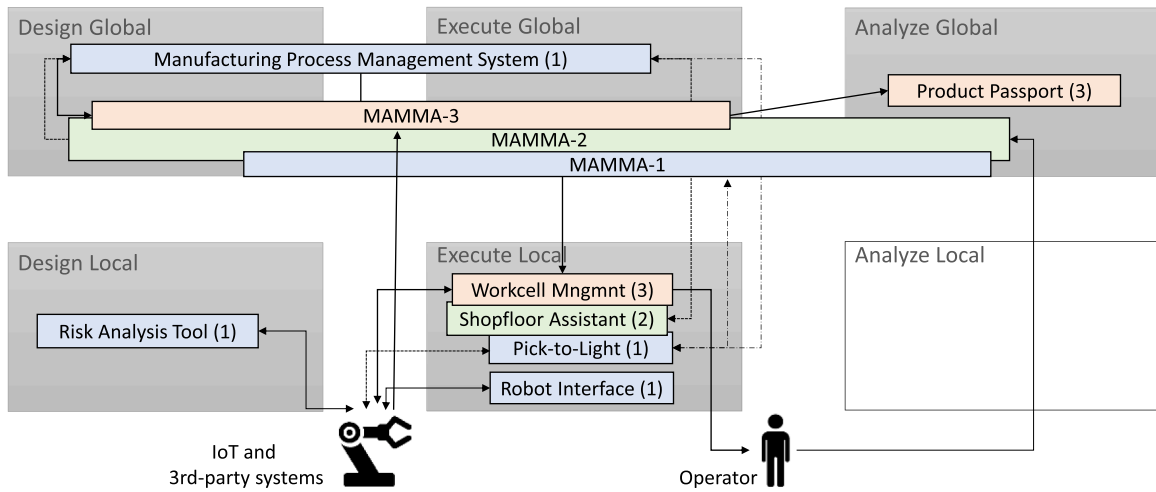


Fig. 20. MATTRESS/Hope-foreman/Sisters initial architecture design.

called MAMMA (Materials Mutation Management), composed of three incremental parts realized in the three iterations (labeled MAMMA-1 to MAMMA-3). MAMMA allows real-time tracking and management of materials on the shop floor. MAMMA is controlled by a manufacturing process management system (MPMS). Apart from the use of pre-existing components (available in the SHOP4CF marketplace), several auxiliary modules were developed in the iterations (shown with functionality-based labels, numbered with the iterations in which they were developed).

The initial architecture was inspired by the initial architecture structure of the SHOP4CF project (corresponding with the bottom part of Fig. 9), but does not comply with guiding principle GP1 as defined in Section 5.1: Fig. 20 clearly shows how components are defined across subsystems, thus violating for example modularity constraints.

The initial architecture has been redesigned using the technical reference architecture presented in Section 5. This has led to the final architecture shown in Fig. 21. In this figure, all instantiated modules are shown. This final architecture clearly has a proper modular design. It shows how the functionality of the MAMMA system is split into

functional modules. The same goes for the functionality of the manufacturing process management system. During the analysis, it also showed that both the manufacturing process management system at the global level and the shop floor assistant at the local level require functionality in the Analyze column. Finally, a data set interface was recognized (indicated by the box icon in the top of Fig. 21), to allow sending packages of data from the Analyze Global and Analyze Local subsystems for external processing. This design decision complies with guiding principle GP5 with respect to external interoperability. The elaboration uses the reference data architecture discussed in Section 5.6, as the design of data sets can be based on the definitions in the *decision data* layer of this data architecture. These extensions complete the architectural specification of the overall system.

Fig. 21 also shows the reuse of components from the marketplace as discussed in Section 4.3. Of the 15 modules in the architecture, 7 modules are based on standard components from the marketplace (with a grayed fill in the figure) and 8 modules are newly developed (with a white fill). The relatively high number of newly developed modules is due to the fact that the application is geared towards product passport

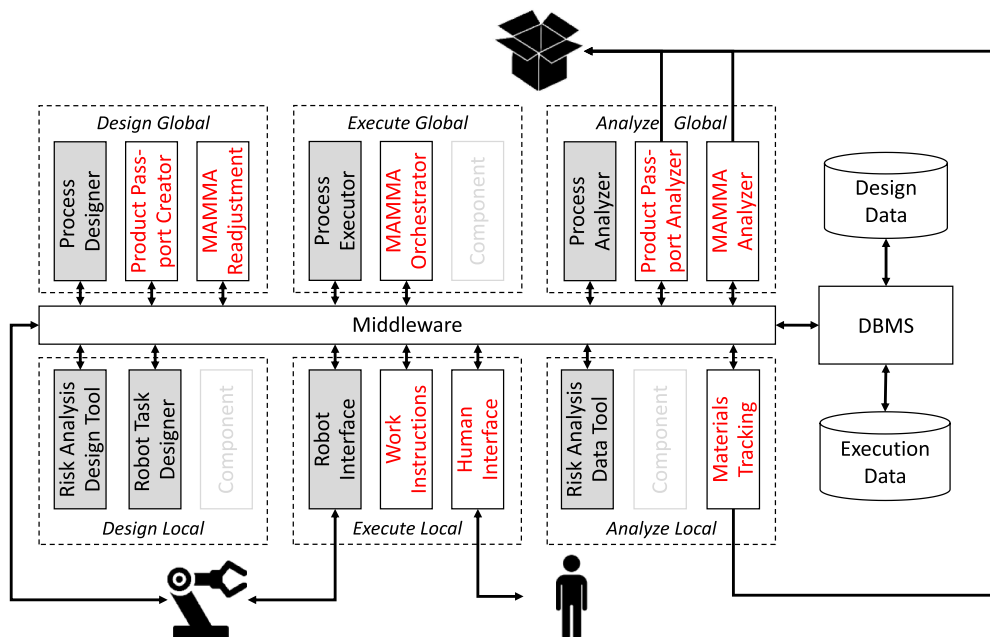


Fig. 21. MATTRESS/Hope-foreman/Sisters final architecture design.

management and materials tracking, functionalities that were not covered by the set of components in the marketplace at the time of system design. Given the component life cycle shown in Fig. 8, this observation can lead to the development of new standard components: given the *suggest* functionality of the marketplace, the development of these components can be triggered by either the manufacturer that owns the system developed with the architecture shown in Fig. 21 or by an integrator assisting in the realization of this system.

The benefits of using the SHOP4CF reference architecture in this case are threefold. Firstly, the number of modules to be designed from scratch was reduced by almost 50 %. The redesigned architecture allows easier mapping to existing components than the original architecture. Secondly, the more modular character of the redesigned architecture makes the solution more future proof (like in the RASP case). Especially the observation that functionality in the design and analysis columns of the solution architecture needs to be split between the global and local levels is important in this case. Thirdly, the use of the reference architecture stressed the fact that adequate attention needs to be paid to the data architecture of the solution and that reference data models can be used as a basis to enhance interoperability (especially with respect to the external interface shown in the top of Fig. 21).

### 7.3. Non-SHOP4CF Case: KUL/OFAA

The Operator-Friendly Agent Allocation (OFAA) project at KU Leuven (KUL) aims at enhancing workforce effectiveness and sustainability in an Industry 5.0 setting by creating an innovative production agent allocation mechanism that takes into account the well-being of the human operators. The mechanism utilizes real-time IoT-based wellbeing metrics, while also considering organizational characteristics (e.g. role, position) and economic process performance objectives.

Human operators often execute heavy physical, mentally demanding or repetitive tasks in a production process. This poses occupational risks to them, such as mental or physical stress, burnout, exhaustion or repetitive strain injury (RSI). This project proposes to take an operator's well-being into account when allocating tasks to agents and to automatically take action if an operator is signaled to be at risk of becoming exhausted or overloaded by re-assigning the tasks to another agent [67]. Biometric information such as heartbeat, sweatiness, skin conductivity, among others, can be monitored by wearable IoT devices to determine the current state of wellbeing of an operator in terms of stress levels, physical fatigue, cognitive (over)load, attention, etc. [68,69]. When tasks have to be (re)allocated the system will take into account this state of wellbeing of each operator and decide which agent is the most suitable resource to execute the task. In that way, fatigued operators may be given less heavy tasks to recuperate, bored or physically strained operators may be given different tasks to improve task variety, heavy tasks

may be re-allocated to a robotic agent, etcetera.

The concept architecture of the KUL/OFAA project is shown in Fig. 22.

To obtain a well-structured architecture for the technical support of the concept of Fig. 22, we map the concept to our technical reference architecture of Fig. 12. This results in the architecture shown in Fig. 23, with again all used modules shown, standard modules shown with a grayed background and newly developed modules with a white background. Based on this mapping, a prototype of this system will be developed as an extension to an existing Manufacturing Process Management System (MPMS), which was realized using the HORSE architecture discussed in Section 2.3.

An evaluation of the workforce well-being will be incorporated into the agent allocation mechanism of the MPMS, based on a well-being assessment of each operator through analysis of the real-time data on the operators' biometric information. As shown in Fig. 23, the real-time biometric data is obtained by the module *Well-Being Data Collector* in the *Execute Local Subsystem*. This module is different in functionality from the *Human Interface* module, which is used to send task specifications to an operator and receive task-based status messages from the operator. The module *Well-Being Analyzer* in the *Analyze Local* subsystem analyzes the well-being of individual operators to obtain information for individual task allocation decisions. The module *Workforce Analyzer* in the *Analyze Global* subsystem aggregates well-being information across multiple operators to obtain information for the design of general, high-level allocation strategies.

The implementation of these modules requires extension of the data architecture as discussed in Section 5.6: the *Resource Definition* class needs to be extended with attributes to store the well-being status of operators – which is reflected in instances in the *Resource Instance* class.

The benefits of the use of the SHOP4CF reference architecture in this case are twofold. Firstly, like for the two realized cases discussed earlier in this section, the design of this new application also shows the potential of reusing existing components through a marketplace: 7 out of 12 projected modules can be based on existing components. Secondly, the redesign on the basis of the reference architecture leads to a more modular architecture that will lead to a more future proof system realization, comparable to the other two discussed cases.

### 7.4. Overall learnings from the SHOP4CF cases

In the SHOP4CF project, we have followed 36 prototype design and implementation efforts for modular software systems in the smart manufacturing domain, two of which are used as illustrations in this section. Most of these efforts have taken place in industrial environments in which the production of software following some 'here-and-now ideas' is more common than the design of well-structured

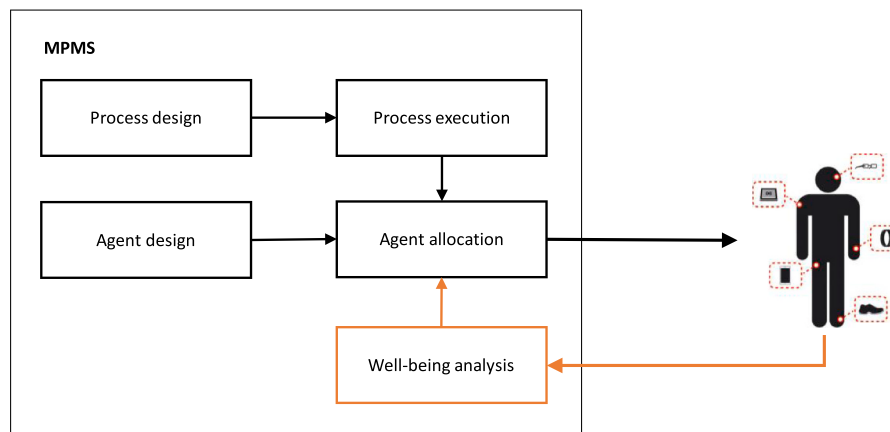


Fig. 22. KUL/OFAA concept architecture.

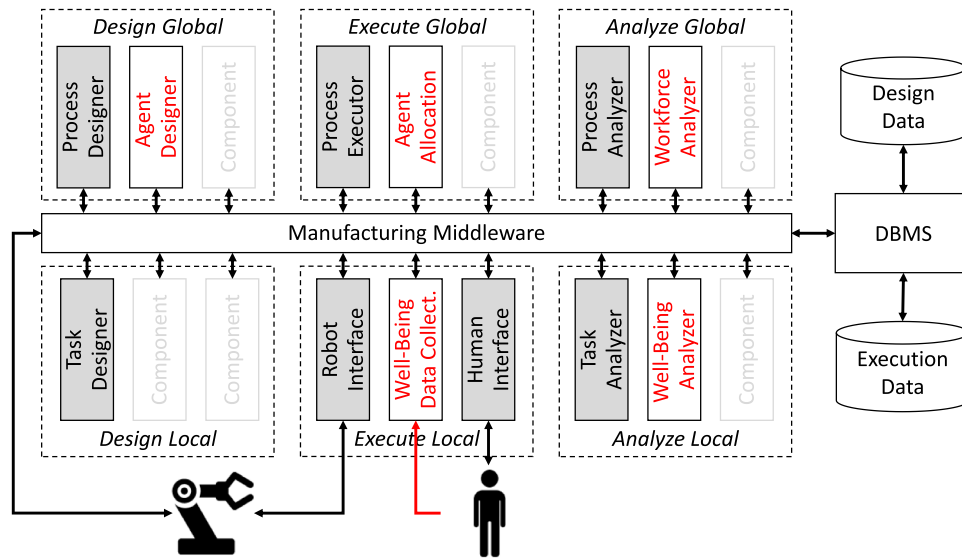


Fig. 23. KUL/OFAA architecture.

architectures that are the basis for the realization of extensible, future-proof systems. With the provisioning of the reference architecture presented in Section 5 and functional elements presented in Section 6, we aimed at steering the efforts towards the architecture-based approach.

In the prototype design processes, however, initial solution design often took place at the software level rather than at the architecture level. This means that software engineers created software solutions and ‘painted’ them in an architecture in an ex-post way – in some cases even violating the design of the reference architecture. This implies a low level of abstraction in system creation, often leading to a lack of flexibility with respect to future evolution and interoperability of created systems. This meant that we needed to provide these design efforts with explicit consultancy regarding architecture-oriented thinking. As illustrated by the two first case studies in this section, this has indeed led to improved architecture designs and hence a better basis for future-proof systems. For example, in both cases, the initial designs included modules spanning multiple subsystems, hence violating the principle of modularity. The redesign based on the reference architecture resolved these issues.

This architecture-based improvement was not a straightforward path, as abstract design is not a common way of working in a very ‘physical’ domain like manufacturing, where software is often still seen as an extension of physical machinery. In hindsight, it probably would have worked better to provide the software designers of the case studies with more ex-ante architecture training to turn them a bit into architects before starting their design efforts. Obviously, the key point here is to convince the designers of the benefits of following an architecture-driven development path.

Even though connected factories were a spearhead of the project, only a few of the 36 use cases in SHOP4CF had a clear role for external interfaces in their architecture and system design. In the two cases described in this section, one is purely internal-oriented and the second includes the first step towards external connections. Consequently, the connected world view (as defined in RAMI 4.0) reflected in our guiding principles GP4 and GP5 (see Section 5.1) and in our reference architecture (see Section 5.5) was hardly used throughout the set of experiments. We see this as a lasting challenge for the European manufacturing industry that still needs explicit addressing.

## 8. Conclusions

The modern economy is changing at still increasing speed and hence implies a high level of flexibility and adaptability for organizations that

want to remain (or become) successful in this economy. This is also valid for the world of manufacturing, despite the fact that many in this world still see it as rather ‘stable’ because of the historically physical orientation of the production infrastructure. Digitization is often seen more as an ‘add-on’ than as the true transformation that is required to become flexible and adaptable. Consequently, in practical solution design in the smart manufacturing world, reference frameworks and architectures are often perceived more as a description (or suggestion) for loosely inspiring system design than as a prescription for solid system architecting. This seriously hinders the design and implementation of robust systems that are maintainable, extensible, portable and interoperable. We have observed this in many case contexts in the early phases of the SHOP4CF project. As we have illustrated in the previous section, the introduction and application of a reference architecture that is based on clear life cycle models and design principles is a first step in addressing this problem.

The importance of using a reference architecture as a design tool to arrive at high-quality digital systems generally has been recognized for a substantial period already [15,17]. The use of a reference architecture contributes to qualities like modularity, extensibility, interoperability, and portability of digital solutions. A reference architecture also can be used as a means of communication in reaching agreement on the scope and functionality of a manufacturing system under design. As such, it is of interest to component developers, system integrators and end users (i. e., the managers of manufacturing plants). The SHOP4CF reference architecture presented in this paper is aimed at contributing to both the quality of smart manufacturing systems and the communication between stakeholders in achieving this quality.

To really anchor better solution design, system developers in smart manufacturing need to be educated in thinking in terms of architectures (and preferably reference architectures). Where this is more commonly accepted practice in more administrative domains (like the financial and service business domains), the manufacturing domain is still heavily lagging behind here. This is probably partly due to the fact that many system developers have a ‘hardware’ background and hence are not so much used to think in abstract structures like architectures. This shows for example in the existence of a gap in many situations between operations technology and information technology, often referred to as the OT-IT dichotomy [19].

Smart manufacturing is largely still an inside-out world: internal capabilities are developed and later related to a value chain. The market developments, though, increasingly demand an outside-in view: value chains or even complex ecosystems define the capabilities that a firm

should have. The execution of the SHOP4CF project, as well as experience in industrial consulting in smart manufacturing, shows that this outside-in view has not yet been generally embraced. The interpretation of the concept of the ‘connected factory’ [19] is too often focused on shop floor connectivity, like in the paper of Calderone [70], and not on business ecosystem connectivity. We hope that the definition of our reference architecture, which includes explicit attention to interfaces to external organizations, can play a positive role in changing this.

In future work, the presented reference architecture can be further extended to provide more functional coverage than the production-focused view of the current version. An interesting extension in the servitization direction of manufacturing is the inclusion of support for product-service systems (PSSs) [71,72]. In our line of design thinking, this extension requires the inclusion of an additional column in the high-level architecture structure shown in Fig. 9, positioned between the *Execute* and *Analyze* columns. This column supports the service execution aspect, where the global level supports the interaction between the service provider (the manufacturer) and the service consumer (the user of the manufactured product), and the local level supports the execution of the service within the service consumer’s business context.

## Funding

This research was partially supported by the SHOP4CF Project funded by the EU Horizon 2020 research and innovation program under grant agreement No. 873087.

## CRedit authorship contribution statement

**Paul Grefen:** Writing – review & editing, Writing – original draft, Visualization, Supervision, Methodology, Investigation, Funding acquisition, Conceptualization. **Zimniewicz Michal:** Writing – review & editing, Writing – original draft, Validation, Software, Methodology, Investigation, Funding acquisition, Conceptualization. **Genessis Perez Rivera:** Writing – review & editing, Writing – original draft, Validation, Investigation. **Irene Vanderfeesten:** Writing – review & editing, Writing – original draft, Validation, Supervision, Project administration, Methodology, Investigation, Funding acquisition, Conceptualization. **Kostas Traganos:** Writing – review & editing, Writing – original draft, Validation, Software, Investigation, Conceptualization. **Pieter Becue:** Writing – review & editing, Writing – original draft, Validation, Project administration, Investigation. **Anders Pedersen:** Writing – review & editing, Writing – original draft, Validation, Project administration, Investigation.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

Many colleagues of the 20 core partners in the SHOP4CF project have contributed to the development of the reference architecture described in this paper. Many colleagues of the additional application partners have contributed to putting the architecture to the test in practice. We thank all of them for their contributions. The research presented in this paper was for a large part supported by the SHOP4CF Project funded by the European Union’s Horizon 2020 research and innovation program under grant agreement No. 873087. The reviewers of the paper are thanked for their constructive feedback, which helped further improve the paper.

## References

- [1] Asadollahi-Yazdi E, et al. Industry 4.0: revolution or evolution? *Am J Oper Res* 2020;10:241–68. <https://doi.org/10.4236/ajor.2020.106014>.
- [2] Grefen P, et al. Dynamic business network process management in instant virtual enterprises. *Comput Ind* 2009;60(2):86–103. <https://doi.org/10.1016/j.compind.2008.06.006>.
- [3] Traganos K, et al. The HORSE framework: a reference architecture for cyber-physical systems in hybrid smart manufacturing. *J Manuf Syst* 2021;61:461–94. <https://doi.org/10.1016/j.jmsy.2021.09.003>.
- [4] Esposito G, Romagnoli G. A reference model for SMEs understanding of Industry 4.0. *IFAC-PapersOnLine* 2021;54(1):510–5. <https://doi.org/10.1016/j.ifacol.2021.08.166>.
- [5] Grefen P, Boultaidakis G. Designing an integrated system for smart industry: the development of the HORSE architecture. 2021 [Independently published]. ISBN: 9798667048640.
- [6] IEC. Enterprise-control system integration - part 1: models and terminology. 2nd ed. Switzerland: The International Electrotechnical Commission (IEC); 2013. ISBN: 9782832207345.
- [7] Hankel M, Rexroth B. The Reference Architectural Model Industrie 4.0 (RAMI 4.0). Germany: German Electrical and Electronic Manufacturers’ Association; 2015. [https://www.zvei.org/fileadmin/user\\_upload/Presse\\_und\\_Medien/Publikationen/2016/januar/GMA\\_Status\\_Report\\_Reference\\_Architecture\\_Model\\_Industrie\\_4.0\\_RAMI\\_4.0/GMA-Status-Report-RAMI-40-July-2015.pdf](https://www.zvei.org/fileadmin/user_upload/Presse_und_Medien/Publikationen/2016/januar/GMA_Status_Report_Reference_Architecture_Model_Industrie_4.0_RAMI_4.0/GMA-Status-Report-RAMI-40-July-2015.pdf).
- [8] Erasmus J, Vanderfeesten I, Traganos K, Grefen P. The case for unified process management in smart manufacturing. Proceedings of the IEEE 22nd international enterprise distributed object computing conference. 2018. p. 218–27. <https://doi.org/10.1109/EDOC.2018.00035>.
- [9] Erasmus J, Vanderfeesten I, Traganos K, Grefen P. Using business process models for the specification of manufacturing operations. *Comput Ind* 2020;123:103297. <https://doi.org/10.1016/j.compind.2020.103297>.
- [10] Lu Y, Xu X, Wang L. Smart manufacturing process and system automation – a critical review of the standards and envisioned scenarios. *J Manuf Syst* 2020;56: 312–25. <https://doi.org/10.1016/j.jmsy.2020.06.010>.
- [11] Mangler J, Ehrendorfer M, Wais B, Rinderle-Ma S. Business process management for efficient manufacturing - sustainability by design. Handbook on business process management and digital transformation. 2024. p. 388–408. <https://doi.org/10.4337/9781802206098>.
- [12] Tao F, Qi Q, Liu A, Kusiak A. Data-driven smart manufacturing. *J Manuf Syst* 2018; 48-C:157–69. <https://doi.org/10.1016/j.jmsy.2018.01.006>.
- [13] Yao X, Zhou J, Lin Y, et al. Smart manufacturing based on cyber-physical systems and beyond. *J Intell Manuf* 2019;30:2805–17. <https://doi.org/10.1007/s10845-017-1384-5>.
- [14] FIWARE. Paving the way for a data-driven industry digitalisation. FIWARE; 2018. [https://www.fiware.org/wp-content/directories/marketing-toolbox/material/FIWAREBrochure\\_SmartIndustry.pdf](https://www.fiware.org/wp-content/directories/marketing-toolbox/material/FIWAREBrochure_SmartIndustry.pdf).
- [15] Bass L, Clements P, Kazman R. *Software architecture in practice*. Addison-Wesley; 2003. ISBN 0321154959.
- [16] Grefen P. *Business information system architecture*. G.DBA; 2025. ISBN 9798306094670.
- [17] Angelov S, Grefen P, Greefhorst D. A framework for analysis and design of software reference architectures. *Inf Softw Technol* 2012;54(4):417–31. <https://doi.org/10.1016/j.infsof.2011.11.009>.
- [18] Treffer A. *Build for change*. Wiley; 2014. ISBN 978-1-118-93028-1.
- [19] Grefen P, et al. Advancing smart manufacturing in Europe: experiences from two decades of research and innovation projects. *Machines* 2022;10(1):45. <https://doi.org/10.3390/machines10010045>.
- [20] Tayler R, Medvidovic N, Dashofy E. *Software architecture: foundations, theory, and practice*. Wiley; 2010. ISBN 978-0-470-16774-8.
- [21] Hollingsworth D. The workflow reference model. Workflow Management Coalition; 1995. <https://wfmc.org/wp-content/uploads/2022/09/tc003v11.pdf>.
- [22] Grefen P, Remmers de Vries R. A reference architecture for workflow management systems. *Data Knowl Eng* 1998;27(1):31–57. [https://doi.org/10.1016/S0169-023X\(97\)00057-8](https://doi.org/10.1016/S0169-023X(97)00057-8).
- [23] Arsanjani A, et al. S3: a service-oriented reference architecture. *IT Prof* 2007;9(3): 10–7. <https://doi.org/10.1109/MITP.2007.53>.
- [24] Pourmirza S, Peters S, Dijkman R, Grefen P. BPMS-RA: a novel reference architecture for business process management systems. *ACM Trans Internet Technol* 2019;19(1):13. <https://doi.org/10.1145/3232677>.
- [25] Lin S. Industrial internet reference architecture (IIRA). Needham MA, USA: Industrial Internet Consortium (IIC); 2015. <https://dspace.mit.edu/bitstream/handle/1721.1/86935/Advisory%20%20IIC%20%20IIRA%20-%20IIC%20Reference%20Architecture.pdf?sequence=190>.
- [26] Moghaddam M, Cadavid MN, Kenley CR, Deshmukh AV. Reference architectures for smart manufacturing: a critical review. *J Manuf Syst* 2018;49:215–25. <https://doi.org/10.1016/j.jmsy.2018.10.006>.
- [27] Li Q, et al. Smart manufacturing standardization: architectures, reference models and standards framework. *Comput Ind* 2018;101:91–106. <https://doi.org/10.1016/j.compind.2018.06.005>.
- [28] Lu Y, Riddick F, Ivezić N. The paradigm shift in smart manufacturing system architecture. Proceedings of the IFIP international conference on advances in production management systems. IFIP; 2016. [https://doi.org/10.1007/978-3-319-51133-7\\_90](https://doi.org/10.1007/978-3-319-51133-7_90).
- [29] IIBM. *Industrie 4.0 architecture*. IBM; 2018.
- [30] Saaksvuori A, Immonen A. *Product lifecycle management*. Springer; 2008. <https://doi.org/10.1007/978-3-540-78172-1>.

- [31] Kletti J. *Manufacturing execution system – MES*. Springer; 2007. ISBN 978-3-540-49743-1.
- [32] Saenz de Ugarte B, Artiba A, Pellerin R. Manufacturing execution system – a literature review. *Prod Plan Control* 2009;20(6):525–39. <https://doi.org/10.1080/09537280902938613>.
- [33] Filos E. Four years of ‘Factories of the Future’ in Europe: achievements and outlook. *Int J Comput Integr Manuf* 2015;30(1):15–22. <https://doi.org/10.1080/0951192X.2015.1044759>.
- [34] Zimniewicz M, et al. SHOP4CF architecture (SHOP4CF Deliverable D3.5, Version 0 to Version 4). SHOP4CF Consortium, Europe. V1 and V2 also online available at <https://portal.effra.eu/project/1949>, V3 at <https://shop4cf.github.io/data-models/download/architecture-v3.pdf>, V4 at <https://shop4cf.github.io/data-models/download/architecture-v4.pdf>; 2020–2023 (all inspected December 2025).
- [35] Perez Rivera, G. et al. (2023). *Prototypes of FSTP Experiments (SHOP4CF Deliverable D5.5)*. SHOP4CF Consortium, Europe. Available at <https://cordis.europa.eu/project/id/873087/results> (inspected December 2025).
- [36] Kazman R, Klein M, Clements P. ATAM: method for architecture evaluation. Report CMU/SEI-2000-TR-004. Carnegie-Mellon University; 2000. <https://doi.org/10.21236/ADA382629>.
- [37] Hevner A, March S, Park J, Ram S. Design science in information systems research. *MIS Q* 2004;28(1):75–105. <https://doi.org/10.2307/25148625>.
- [38] Greefhorst D, Proper E. *Architecture principles*. Berlin: Springer; 2011. <https://doi.org/10.1007/978-3-642-20279-7>.
- [39] Gyulai D, Bergmann J, Gallina V, Gaal A. Towards a connected factory: shop-floor data analytics in cyber-physical environments. *Procedia CIRP* 2019;86:37–42. <https://doi.org/10.1016/j.procir.2020.01.016>.
- [40] Aalst W van der. *Process mining: data science in action*. 2nd ed. Springer; 2016. <https://doi.org/10.1007/978-3-662-49851-4>.
- [41] Mayr A, Herm L, Wanner J, Janiesch C. Applications and challenges of task mining: a literature review. Proceedings of the European conference on information systems. 2022. [https://aisel.aisnet.org/ecis2022\\_rip/55/](https://aisel.aisnet.org/ecis2022_rip/55/).
- [42] Mehandjiev N, Grefen P, editors. *Dynamic business process formation for instant virtual enterprises*. Springer; 2010. <https://doi.org/10.1007/978-1-84882-691-5>.
- [43] Magal S, Word J. *Integrated business processes with ERP systems*. Wiley; 2011. ISBN: 978-0-470-47844-8, <https://dl.acm.org/doi/abs/10.5555/2011876>.
- [44] Sagegg O, Alfnes E. *ERP systems for manufacturing supply chains: applications, configuration, and performance*. Auerbach Publications, CRC Press; 2020. ISBN 9781032474762, <https://www.routledge.com/ERP-Systems-for-Manufacturing-Supply-Chains-Applications-Configuration-and-Performance/Sagegg-Alfnes/p/book/9781032474762>.
- [45] Alexopoulos K, Sipsas K, Xanthakis E, Makris S, Mourtzis D. An Industrial Internet of Things based platform for context-aware information services in manufacturing. *Int J Comput Integr Manuf* 2018;31(11):1111–23. <https://doi.org/10.1080/0951192X.2018.1500716>.
- [46] Arnold C, Kiel D, Voigt K. How the Industrial Internet of Things changes business models in different manufacturing industries. *Int J Innov Manag* 2016;20(8):1640015. <https://doi.org/10.1142/S1363919616400156>.
- [47] Leng J, et al. Resilient manufacturing: a review of disruptions, assessment, and pathways. *J Manuf Syst* 2025;79:563–83. <https://doi.org/10.1016/j.jmsy.2025.02.006>.
- [48] Comuzzi M, Grefen P, Meroni G. *Blockchain for business*. Routledge; 2023. ISBN 9781032342467, <https://www.routledge.com/Blockchain-for-Business-IT-Principles-into-Practice/Comuzzi-Grefen-Meroni/p/book/9781032342467>.
- [49] Steinbuss S, editor. *Reference architecture model version 3.0*. Berlin, Germany: International Data Spaces Association; 2019. <https://internationaldataspaces.org/wp-content/uploads/IDS-Reference-Architecture-Model-3.0-2019.pdf>.
- [50] Hastings N, Yeh C. Bill of manufacture. *Prod Inventory Manag J* 1992;33(4):27. <https://www.scopus.com/pages/publications/0027111884>.
- [51] Onuralp Gökcalp M, et al. Data-driven manufacturing: an assessment model for data science maturity. *J Manuf Syst* 2021;60:527–46. <https://doi.org/10.1016/j.jmsy.2021.07.011>.
- [52] Zachman J. A framework for information systems architecture. *IBM Syst J* 1987;26(3):276–92. <https://doi.org/10.1147/sj.263.0276>.
- [53] Zachman J. Evolution of the Zachman framework. 2009. (<https://www.zachman.com/resource/ea-articles/54-the-zachman-framework-evolution-by-john-p-zachman>).
- [54] Sandkuhl K, Stirna J, Persson A, Wißotzki M. *Enterprise modeling*. Heidelberg: Springer; 2014. p. 24–75. ISBN 9783662524459, <https://link.springer.com/book/10.1007/978-3-662-43725-4>.
- [55] Kumaran S, Liu R, Wu FY. On the duality of information-centric and activity-centric models of business processes. Proceedings of the conference on advanced information systems engineering. Berlin, Heidelberg: Springer; 2008. [https://doi.org/10.1007/978-3-540-69534-9\\_3](https://doi.org/10.1007/978-3-540-69534-9_3).
- [56] Kumar A. *Business process management*. Routledge; 2018. ISBN 9781138181854.
- [57] Reijers HA, et al. Evaluating data-centric process approaches: does the human factor factor in? *Softw Syst Model* 2017;16:649–62. <https://doi.org/10.1007/s10270-015-0491-z>.
- [58] Sun SX, Zhao JL, Nunamaker JF, Sheng ORL. Formulating the data-flow perspective for business process management. *Inf Syst Res* 2006;17(4):374–91. <https://doi.org/10.1287/isre.1060.0105>.
- [59] Vanderfeesten I, et al. Developing process execution support for high-tech manufacturing processes. In: Lübke D, Pautasso C, editors. *Empirical studies on the development of executable business processes*. Springer; 2019. [https://doi.org/10.1007/978-3-030-17666-2\\_6](https://doi.org/10.1007/978-3-030-17666-2_6).
- [60] Nguyen T, et al. Wearable technology for smart manufacturing in Industry 5.0. In: Tran KP, editor. *Artificial intelligence for smart manufacturing*. Springer; 2023. [https://doi.org/10.1007/978-3-031-30510-8\\_11](https://doi.org/10.1007/978-3-031-30510-8_11).
- [61] Rozhok A, Abate R, Manoli E, Nele L. A review of recent advanced applications in smart manufacturing systems. *J Manuf Mater Process* 2025;10(1):1. <https://doi.org/10.3390/jmmp10010001>.
- [62] Sowdaboina P, Chakraborti S, Sripada S. Learning to summarize time series data. In: Gelbukh A, editor. *Computational linguistics and intelligent text processing*. Springer; 2014. [https://doi.org/10.1007/978-3-642-54906-9\\_42](https://doi.org/10.1007/978-3-642-54906-9_42).
- [63] Leng J, et al. Diffusion model-driven smart design and manufacturing: prospects and challenges. *J Manuf Syst* 2025;82:561–77. <https://doi.org/10.1016/j.jmsy.2025.07.011>.
- [64] Grefen P, et al. Towards customer outcome management in smart manufacturing. *Machines* 2023;11(6):636. <https://doi.org/10.3390/machines11060636>.
- [65] Semianiaka, N., Silina, E. (2012). *The role of global data identification standards for supply chain visibility: the case of GS1*. Available at <https://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-107888> (inspected December 2025).
- [66] Rinderle-Ma S, Mangler J. Process automation and process mining in manufacturing processes. Proceedings of the international conference on business process management. Springer; 2021. [https://doi.org/10.1007/978-3-030-85469-0\\_1](https://doi.org/10.1007/978-3-030-85469-0_1).
- [67] Wyffels M. Integrating real-time worker wellbeing into IoT-enhanced business process orchestration for Industry 5.0. Proceedings of the international conference on advanced information systems engineering. Springer; 2025. [https://doi.org/10.1007/978-3-031-94590-8\\_31](https://doi.org/10.1007/978-3-031-94590-8_31).
- [68] Loizaga E, Eyam A, Bastida L, Lastra J. A comprehensive study of human factors, sensory principles, and commercial solutions for future human-centered working operations in Industry 5.0. *IEEE Access* 2023;53806–29. <https://doi.org/10.1109/ACCESS.2023.3280071>.
- [69] Bavaresco R, Arruda H, Rocha E, Barbosa J, Li GP. Internet of things and occupational well-being in Industry 4.0: a systematic mapping study and taxonomy. *Comput Ind Eng* 2021;161. <https://doi.org/10.1016/j.cie.2021.107670>.
- [70] Calderone, L. (2018). *The Connected Factory*. Manufacturing Tomorrow. Available online: <https://www.manufacturingtomorrow.com/article/2018/01/the-connected-factory/10819> (inspected April 2025).
- [71] Alexopoulos K, Koukas S, Boli N, Mourtzis D. Architecture and development of an Industrial Internet of Things framework for realizing services in Industrial Product Service Systems. *Procedia CIRP* 2018;72:880–5. <https://doi.org/10.1016/j.procir.2018.03.152>.
- [72] Zheng P, Lin T, Chen C, Xu X. A systematic design approach for service innovation of smart product-service systems. *J Clean Prod* 2018;201:657–67. <https://doi.org/10.1016/j.jclepro.2018.08.101>.