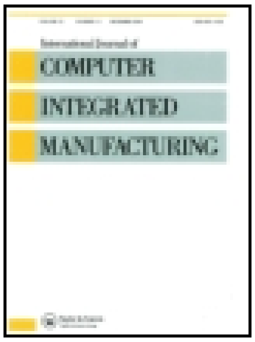

International Journal of Computer Integrated Manufacturing

ISSN: 0951-192X (Print) 1362-3052 (Online) Journal homepage: https://www.tandfonline.com/loi/tcim20

An open source approach to the design and implementation of Digital Twins for Smart Manufacturing

Violeta Damjanovic-Behrendt & Wernher Behrendt

To cite this article: Violeta Damjanovic-Behrendt & Wernher Behrendt (2019): An open source approach to the design and implementation of Digital Twins for Smart Manufacturing, International Journal of Computer Integrated Manufacturing, DOI: 10.1080/0951192X.2019.1599436

To link to this article: https://doi.org/10.1080/0951192X.2019.1599436

Published online: 13 Apr 2019.

INTERNATIONAL JOURNAL OF COMPUTER INTEGRATED MANUFACTURING
https://doi.org/10.1080/0951192X.2019.1599436

Taylor & Francis
Taylor & Francis Group

ARTICLE

An open source approach to the design and implementation of Digital Twins for Smart Manufacturing

Violeta Damjanovic-Behrendt and Wernher Behrendt

The IoT Group, Salzburg Research, Salzburg, Austria

ABSTRACT
This paper discusses the design of a Digital Twin (DT) demonstrator for Smart Manufacturing, following an open source approach for implementation. Open source technology can comprise of software, hardware and hybrid solutions that nowadays drive Smart Manufacturing. The major potential of open source technology in Smart Manufacturing lies in enabling interoperability and in reducing the capital costs of designing and implementing new manufacturing solutions. After presenting our motivation to adopt an open source approach for the design of a DT demonstrator, we identify the major implementation requirements of Smart Cyber Physical Systems (CPSs) and DTs. A conceptualisation of the core components of a DT demonstrator is provided and three technology building blocks for the realisation of a DT have been identified. These technology building blocks include components for the management of data, models and services. From the conceptual model of the DT demonstrator, we derived a high-level micro-services architecture and provided a case study infrastructure for the implementation of the DT demonstrator based on available open source technologies. The paper closes with research questions to be addressed in the future.

ARTICLE HISTORY
Received 31 May 2018
Accepted 13 March 2019

KEYWORDS
Digital Twin; Smart CPS; open source technology; micro-services architecture

1. Introduction

Open source software provides the key building blocks for interoperability and flexibility of Smart Manufacturing solutions (IoT Eclipse.org 2017). When using permissive open source licences, the technology is freely redistributable and modifiable, supporting manufacturers in combining older equipment with modern sensor-based machines and tools from different vendors. Technical scalability and computational power for data analytics are major requirements of manufacturing software solutions and are today dominated by open source software. Similarly, open source hardware supports faster prototyping and customisation of reprogrammable components of Cyber Physical Systems (CPSs), which helps manufacturers to accelerate the design and improve interoperation across actual lifecycle processes. Enhanced interoperation of cost-effective manufacturing solutions, based on open source technology, also reduces fragmentation of supply chains and facilitates collaboration among numerous manufacturing enterprises.

Many open source communities like the Apache Software Foundation, the Linux Foundation, the Eclipse IoT and others, have become valuable technology suppliers to the Smart Manufacturing software industry. For example, the Eclipse IoT Working Group has 28 projects that target general Internet of Things (IoT) solutions, with some of them being applicable to manufacturing (IoT Eclipse.org 2017). Since open source technology often uses permissive licences and is royalty-free, making use of it reduces the costs of creating new solutions and enables free participation in the future development and quality control of Smart Manufacturing technology.

In parallel with the continuous advancement of open source technology, the concepts of Smart CPSs and DTs are undergoing rapid changes addressing a plethora of challenges, such as multitenancy, data sharing, cybersecurity, governance models for manufacturing platforms, data stream capturing along life-cycle models. While research is still looking for suitable architectures, the next wave of desired features for Smart Manufacturing is already being defined: autonomous decision making, context- and situation-aware controls, self-adaptation and more.

Yet at present, DT platforms are built as closed systems, thus limiting the overall advantages of Smart Manufacturing. Hence, the major motivation for this paper is to design a flexible, open source solution for DTs and make it accessible to a wider industrial and research audience. The paper introduces a functional architecture of a DT and explores the potentials of available open source tools and services to be composed into such a demonstrator.

The paper is organised as follows. Section 2 presents the state of practise in Industry 4.0 and Smart Manufacturing, and discusses current work related to major technology enablers for CPS, Smart CPSs and DTs. Section 3 defines the conceptual model of a DT demonstrator and describes its core building blocks: Data Manager, Models Manager and Services Manager. An open source software technology for the DT demonstrator, addressing the high-level requirements of the presented conceptual model of a DT, is elaborated in Section 4. In Section 5, we firstly explain our motivation to implement DTs as micro-services. Secondly, we provide a mapping of the DT conceptual architecture to the IIRA and RAMI 4.0 reference models.

CONTACT Violeta Damjanovic-Behrendt ✉ violeta.damjanovic@salzburgresearch.at

Thirdly, we present the design of a DT architecture that follows new design patterns based on micro-services. Section 6 presents a possible infrastructure for implementing a DT demonstrator. Section 7 gives lines of future research and Section 8 concludes the paper.

2. Related work

This section explores related work and current practises in Industry 4.0 and Smart Manufacturing. The section presents the major technology enablers for Smart CPSs and DTs, together with core reference models for interoperation and standardisation in both domains, Industry 4.0 and Smart Manufacturing. To illustrate the current state of manufacturing practises, this section overviews several available examples of open source and commercially designed DTs.

2.1. State of practise in Industry 4.0 and Smart Manufacturing

Manufacturing industry is moving towards a new paradigm, with the objective to increase productivity, efficiency and competitiveness through Industry 4.0 and Smart Manufacturing. The idea of Industry 4.0 came from the German government promoting a 'High Tech Strategy 2020 Action Plan' in 2013 (MacDougall 2014), while the idea of Smart Manufacturing evolved from several sources, including: (i) computer integrated manufacturing (CIM) in the 1980s, (ii) Reconfigurable Manufacturing Systems (RMSs) (Koren et al. 1999; Koren and Shpitalni 2010), (iii) the Smart Factory initiative proposed in Zuehlke (2010) and (iv) the Ubiquitous Factory concept from Yoon, Shin, and Suh (2012). Today, both strands of research and development follow common manufacturing challenges related to: *flexibile and modular* production lifecycle and processes to address changing manufacturing requirements, *interoperability* to support data exchange between digital entities and their physical counterparts, *optimisation* of processes and infrastructures based on Smart CPSs and DTs, as well as the ever present *cybersecurity and privacy challenges*.

Industry 4.0 and Smart Manufacturing rely on technologies such as IoT, Web of Things (WoT), cloud computing, edge computing, big data and analytics, smart sensors, CPSs, DTs and artificial intelligence (AI) (Wang et al. 2016) (Cimini, Pinto, and Cavalieri 2017). For example, the role of IoT in digital manufacturing is to create and enable the collection of real-time sensor data that can be exchanged through the Web (Sarma, Brock, and Ashton 2000). Edge and cloud computing technologies further support systems for advanced analysis and correlation of data; AI technologies enable data mining and the creation of added value through knowledge discovery. Big data technologies provide systematic analyses of a variety of data generated along the entire product lifecycle, and improve productivity of manufacturing systems through rapid decision making (Davis et al. 2012; Lee, Kao, and Yang 2014). CPS is another key technology that adds intelligence to traditional production processes (Lee, Bagheri, and Kao 2015; Jazdi 2014). CPS integrates computational paradigms with the physical processes (Lee 2008) and supports manufacturing capabilities such as reliability, interoperability, predictability and tracking. CPSs are expected to play a major role in the design and implementation of future software systems with new capabilities, as noted in Baheti and Gill (2011) and Rajkumar et al. (2010). Monostori (2014) defines a Cyber Physical Production System (CPPS) as an interconnected system operating *'…across all levels of production, from processes through machines up to production and logistics networks'*. The term CPPS is often used as a synonym for the Smart Factory, emphasising the scalable and modular structure of Smart Manufacturing applications (Weyer et al. 2016). Further advancements of CPSs are known as Smart CPSs, which are complex engineering systems that integrate heterogeneous hardware and software technologies through various analytics and decision-making mechanisms (Horváth and Gerritsen 2012). Tavčar and Horváth (2018) summarise the core advancements of Smart CPSs in comparison to CPSs. On the following scale, the authors consider only CPSs of Level 2 and above as Smart CPSs:

- Level 1. CPS does not change throughout the system life span. It has conventional control mechanisms and can regulate parameters to a known degree.
- Level 2. CPS is designed for alternative modes of control and for selection of the optimal control mode at run-time.
- Level 3. A self-learning CPS with the ability to adapt its pre-defined control algorithms during operation.
- Level 4. A CPS that is designed to react intelligently to previously not anticipated changes.

The evolution of microchip, sensor and WoT technologies has opened the way for tracking smart products along their lifecycle phases, analysing the acquired data and communicating their production and operating conditions (Schleich et al. 2017). This technology evolution shifted the concept of Digital Twins from the aerospace industry to Smart Manufacturing (Rios et al. 2015). The concept of twins in aircraft industries has been used to support the optimisation and validation technology of aircraft systems, based on the integration of sensor data, maintenance data and available historical/fleet data (Shafto et al. 2010). In 2002, Grieves coined the term DT that subsequently evolved from 'conceptual ideal for product lifecycle management (PLM)', 'the mirrored space model', 'the information mirroring model', to today's slightly varying definitions of a DT. For example, Grieves (2014) and Grieves and Vickers (2017) define a DT as 'a set of virtual information constructs that fully describes a potential or actual physical, manufactured product from the micro-atomic level to the macro geometrical level'. In addition, a DT is also defined as a 'digital counterpart of a physical product' (Rios et al. 2015) and the term is also used for simulation and prediction of future states of the physical product (e.g. Gabor, Belzner, and Kiermeier 2016). Haag and Anderl (2018) define a DT as a comprehensive digital representation of an individual product, its properties, conditions and behaviour. The core functionality of a DT is to support design tasks and to validate system properties through multi-domain and multi-level simulations along lifecycle phases, including operational support (Boschert and Rosen 2016). Negri, Fumagalli, and Macchi (2017) give a comprehensive overview of DT definitions currently available in the literature.

2.1.1. Industry 4.0 reference architectures

At present, there exist two core reference models for interoperation and standardisation in Industry 4.0 and Smart Manufacturing: the Reference Architecture Model for Industry 4.0 (RAMI 4.0) and the Industrial Internet Reference Architecture (IIRA). The authors in ZVEI (2015) present the RAMI 4.0 architecture as a model in a three-dimensional space: the two horizontal axes represent (i) the value chain and the lifecycle, and (ii) the different hierarchies of a production system (i.e. products, field devices, control devices, station, work centres, enterprise, connected world), while the vertical axis represents the layers that describe the physical world (asset), the integration of software and hardware components, communication capabilities, information creation through data, functional properties and business processes.

The Industrial Internet Consortium (2017) presents the IIRA architecture model that maps the five functional domains (e.g. control, operation, information, application and business) against system characteristics (e.g. safety, security, privacy, resilience, scalability, reliability) and cross-cutting functions (e.g. connectivity, distributed data management, industrial analytics, intelligent control).

2.1.2. A view of current commercial and open source Digital Twin solutions

The examples of commercial software solutions that implement industrial DT technology include General Electric, PTC Windchill, Dassault Systèmes, DXC, Siemens Simcenter, Microsoft Azure Digital Twins and Seebo Digital Twin.

- General electric (GE) developed the DT of a jet engine that enables the configuration of individual wind turbines, prior to their procurement and construction. Each virtual turbine is then fed with data from its physical equivalent. GE's DT is based on the Predix platform (www.predix.com) that delivers capabilities such as asset connectivity, edge technologies, analytics and machine learning (ML), big data processing, asset performance management (APM) and implements asset-centric DTs (Predix, 2018). Apart DT, GE's Digital Twin Starter Kit is an open source toolset used to teach software engineers how to build DTs. The DT Starter Kit is based on the Predix infrastructure that enables constructing and hosting DTs (GitHub page: https://github.com/DigitalTwin).
- PTC Windchill (2018) has developed a Smart PLM software called Windchill with failure reporting, analysis and a notification system for corrective actions.
- Dassault Systèmes (DS) has built an aerospace—and defence—manufacturing management software called *Build to Operate*, which enables monitoring, controlling and validation of all aspects of manufacturing operations (Grieves 2014).
- DXC has developed a DT for predicting performances of hybrid cars, before committing the changes in the car manufacturing process (Overton and Brigham 2017). DXC has partnered with Microsoft to build ML solutions on an industrial scale, and used the Microsoft Cortana Intelligence Suite to run the DT, through continuous simulation of new ways for creating hybrid cars.
- The authors in Siemens Simcenter (2017) present the Simcenter 3D for DT. The Simcenter software distinguishes between (i) a product DT that allows users to virtually execute new designs and simulate the effects of the changes in the digital system, (ii) a production DT that is used to validate the effectiveness of a manufacturing process created for the factory floor assets and (iii) a performance DT that processes big data from industrial IoT (IIoT) products in smart plants in order to improve product and production systems efficiency.
- Bosch IoT (https://www.bosch-iot-suite.com/) is a commercial product that uses the DT approach for asset management, and sharing device data and functionality across applications.
- Microsoft Azure Digital Twin (https://docs.microsoft.com/en-gb/azure/digital-twins/) is one of the recent Microsoft Azure cloud services used to create comprehensive models of the physical environment. In the manufacturing context, it can be used for predicting maintenance needs for a factory and for analysing real-time energy requirements for an electrical grid.
- Seebo Digital Twin software (https://www.seebo.com/digital-twin-software/) provides a visual modelling tool for the graphical design of digital replicas of production line processes and assets. A functional DT prototype can be generated directly from the designed DT model. By using the Seebo IoT Simulator (https://www.seebo.com/iot-simulation/), the use cases, data flow human machine interfaces (HMIs) and predictive quality systems on the manufacturing assets, can be validated.

The open source community is at the forefront of creating software and hardware solutions for IoT experimentation in Industry 4.0 and Smart Manufacturing. However, there are currently only a few open source DTs solutions available, among them Eclipse Ditto and CPS Twinning.

- Eclipse Ditto is an open source software solution that enables the design of DTs in a form of IoT development patterns. Bosch IoT Things is an example of the commercial product that is based on open source Eclipse Ditto (GitHub page: https://github.com/eclipse/ditto).
- CPS Twinning is a framework used to automatically generate virtual environments for DTs from an AutomationML (AML)-based specification (GitHub page: https://github.com/sbaresearch/cps-twinning).

2.1.3. Conclusion on practises in Industry 4.0 and Smart Manufacturing

With the evolution of traditional industrial systems towards Industry 4.0 and Smart Manufacturing (Kang et al. 2016), it is expected that manufacturing systems and processes become more adaptive and flexible through integrated automated decision-making mechanisms, self-awareness and self-optimisation features (MöLler 2016). New virtual models of real factory settings, known as DTs, are expected to be designed to:

- ensure information exchange throughout the entire lifecycle (Rosen et al. 2015),
- enable virtualisation of manufacturing systems (Schluse and Rossmann 2016),
- automate decision making and system behaviour-based predictions (Kraft 2016).

Currently available DTs are either fully commercial solution, or built as open source-based commercial solutions (i.e. Bosch IoT Suite based on Eclipse Ditto).

2.2. Technology enablers for Digital Twins

The ultimate objective of DTs is to improve the operation and efficiency of manufacturing assets, reduce costs through forecasting and predictions of future states and support advanced decision making through the entire manufacturing lifecycle. Hence, the following elements need to be considered for the design of DT (Oracle 2017):

- Asset modelling,
- Predictive analytics and decision-making methods, and
- Lifecycle-oriented knowledge base with historical and real-time sensor data.

2.2.1. Asset modelling

Asset modelling is about architecting DT through designing the structure of its assets (physical things) and components, measurable parameters and manufacturing information about the assets (e.g. manufacturing date, maintenance history) (Kucera, Aanenson, and Benson 2017). Asset modelling adds value to connected sensor data and contributes to a range of new insights, e.g. obtaining information on asset health through sensors, which can be performed through inference, correlation and transformation of measured sensor values and asset states, conditions and maintenance records. It may also provide different presentation (visualisation) forms for different user groups (stakeholders), e.g. one group of users may require the insight in only operational data, while others are more focused on individual devices. Adding information such as metadata, nearby environmental conditions, maintenance data, service history, configuration and production data, enterprise web services, etc., contributes to a rich representation of the physical things and further augments the DT.

2.2.2. Predictive analytics for Digital Twins

Predictive analytics comprise a variety of techniques for calculating future outcomes based on historical and real-time data. It seeks to uncover patterns and capture relationships in data through techniques such as (Gandomi and Haider 2015):

- moving averages—discovers the historical patterns in the outcome variable(s) and extrapolate them to the future, and
- linear regression—captures the interdependencies between outcome variable(s) and explanatory variables, and use them to create predictions.

Based on the underlying techniques, predictive analytics can be categorised into two groups (Gandomi and Haider 2015):

- regression techniques (e.g. multinomial logit models) and
- ML techniques (e.g. neural networks, supervised learning, unsupervised learning, reinforcement learning (Sutton and Barto 2012)).

Predictive analytics techniques are primarily based on statistical methods, which often, when applied to massive data of DTs, do not scale up in terms of computational efficiency. Big data is characterised by factors such as heterogeneity, noise accumulation, spurious correlations, incidental endogeneity (Fan, Han, and Liu 2014) and requires new statistical techniques to gain insights from predictive models. Specifically, in Cloud computing, some relevant approaches are based on task resource consumption patterns (Mishra et al. 2010) and the usage of storage systems (Aggarwal, Phadke, and Bhandarkar 2010). The analysis of behaviour patterns and derived models has been discussed in (Bahga and Madisetti 2011; Chen et al. 2010; Smith and Sommerville 2011). Yang et al. (2012) presents the principal component analysis (PCA) technique used to retrieve relations between configuration and resource usage and performances in Cloud computing.

The core predictive models for behavioural analysis can be also classified in two groups: location detection techniques and temporal behaviour analysis of time series. A variety of available location detection technologies leads to the massive accumulation of online data about users/assets location and their activity/usage histories. Such data are used for mining knowledge in applications ranging from location-based recommendation systems to applications for tracking user/asset movements and activities. For example, pattern mining of GPS readings is often designed to identify specific patterns in a users' movement and behaviour (Geng, Arimura, and Uno 2012); the k-Means clustering algorithm is used to learn user's significant location and daily routines from his location history (Ashbrook and Starner 2002, 2003); pattern mining from very large historical spatio-temporal dataset (Tsoukatos and Gunopulos 2001); mining location patterns using Hidden Markov Models that can further feed frequent pattern mining methods, as presented in Qiu and Bandara (2015).

Regarding temporal behaviour analysis of time series, the most common approaches to modelling time series are: trend, seasonal, residual decomposition, frequency-based methods, auto-regressive methods (AR), moving average (MA). The Conditional Restricted Boltzmann Machine (CRBM) is a probabilistic model for time series used to solve a range of problems, from classification tasks to collaborative filtering and modelling of the motion capture (Mnih, Larochelle, and Hinton 2011; Taylor and Hinton 2009).

2.2.3. Lifecycle-oriented knowledge base of Digital Twins

The DT knowledge base has to cover a wide range of diverse data: asset lifecycle data (e.g. location-based and time-series sensor data), data derived from analytics and decision-making algorithms, expert data, regulatory data, historical data. The DT knowledge base is often augmented by adding data from a variety of third-party data sources, e.g. asset maintenance history from an enterprise resource planning (ERP) system, account data from a customer relationship management (CRM) system, environmental data, etc. One of the critical prerequisites to the DT knowledge base creation is to have a proper data integration platform and infrastructure in place, enabling the integration of multiple data streams through standards and frameworks (Oracle 2017).

According to the size of a DT knowledge base and its maturity level, the authors in (Kucera, Aanenson, and Benson 2017) differentiate among:

- a partial DT, with a small number of data sources that can be combined to infer further data (derivative data),
- a clone DT, with a larger amount of meaningful and measurable data sources and
- an augmented DT, that enhances connected asset data with derivative data and correlated data obtained from analytics tools.

A partial DT is a set of simplistic device models that could be implemented as JSON documents with a set of observed and reported attributes (e.g. speed of a machine) and a set of desired values (e.g. an application is setting the speed of a machine) that can be correlated to detect operational abnormalities and instantly generate alerts. A clone DT is typically what is needed in industry: it is built on top of the product design and manufacturing information, and reflects its physical properties and uses real-time data (Kucera, Aanenson, and Benson 2017).

2.2.4. Conclusion

This section summarises core technology enablers for the design and implementation of DTs, which include asset modelling, predictive analytics techniques and a lifecycle-oriented knowledge base with historical and real-time sensor data. According to the size of the lifecycle-oriented knowledge base of DTs, they could be designed as partial, clone or augmented DTs based on the required accuracy of the domains to be represented by DTs.

3. Conceptual view of a Digital Twin demonstrator in Smart Manufacturing

One of the desired features of DT technology in Smart Manufacturing is the ability to accurately simulate, analyse and predict events and situations from the manufacturing ecosystem. In order to do so, the high-level DT technology requirements can be summarised as follows:

- Firstly, the DT technology requires a variety of data to be collected, analysed and 'mirrored' in the form of decisions and feedbacks sent from virtual space back to the manufacturing ecosystem.
- Secondly, the DT technology requires a collection of models created with the aim to describe manufacturing lifecycle phases, e.g. Kiritsis (2011). The DT models vary from complex models, to simplified ones that include only the most relevant simulation and predictive models, and operational data that evolve throughout various lifecycle phases (Boschert and Rosen 2016). The DT models exist either as computational models (e.g. statistical packages for ML, analytics, optimisation) or representational models (e.g. semantic data models, NoSQL, relational data models, relational derivatives, rule engines).
- Thirdly, the DT technology needs to be equipped with a collection of services to effectively monitor and simulate the physical world and perform computations leading to decisions and feedbacks.

To reduce the complexity of the DTs technology and to keep a strong focus on the functionality of DTs, we designed the conceptual model along the above three design rationales. As illustrated in Figure 1, the Virtualisation Manager is at the heart of a DT and comprises of three building blocks: the Data Manager, the Models Manager and the Services Manager. Apart from the Virtualisation Manager, the DT contains another four management components: the Monitoring Manager, the Decision-Making Manager, the Simulation Manager and the Interoperation Manager. The Monitoring Manager provides the connectivity between the Virtualisation Manager and the assets of the factory shop floor. The Decision-Making Manager is responsible for the presentation formats of feedbacks created through analytics services of the Virtualisation Manager. The Simulation Manager provides the simulation formats based on visualisation dashboard services of the Virtualisation Manager.

The rest of this section provides a detailed description of the three core building blocks of the Virtualisation Manager.

3.1 Data Manager

The Data Manager of the DT demonstrator includes the data acquisition and the data analytics components (see Figure 1).

- Data acquisition: In DTs, the collected data often comprise of real-time (or near real-time) sensor data, expert knowledge data, historical data and inferred data that are generated along the entire product lifecycle and aggregated in big data sets, data integrated from other enterprise systems and third-party systems. For example, the data collected from the design lifecycle phase in Smart Manufacturing include data for model building, model function, model design, computer-aided design (CAD), configuration and parameter optimisation, structure, mechanics, size, material, history, predictions, simulations, processes, environment, faults, redesign activities, customers reviews and feedback (Tao et al. 2017, 20199). The data collected from the manufacturing lifecycle encompass manufacturing instructions, casting and moulding data, computer-aided manufacturing (CAM) planning data, and more.

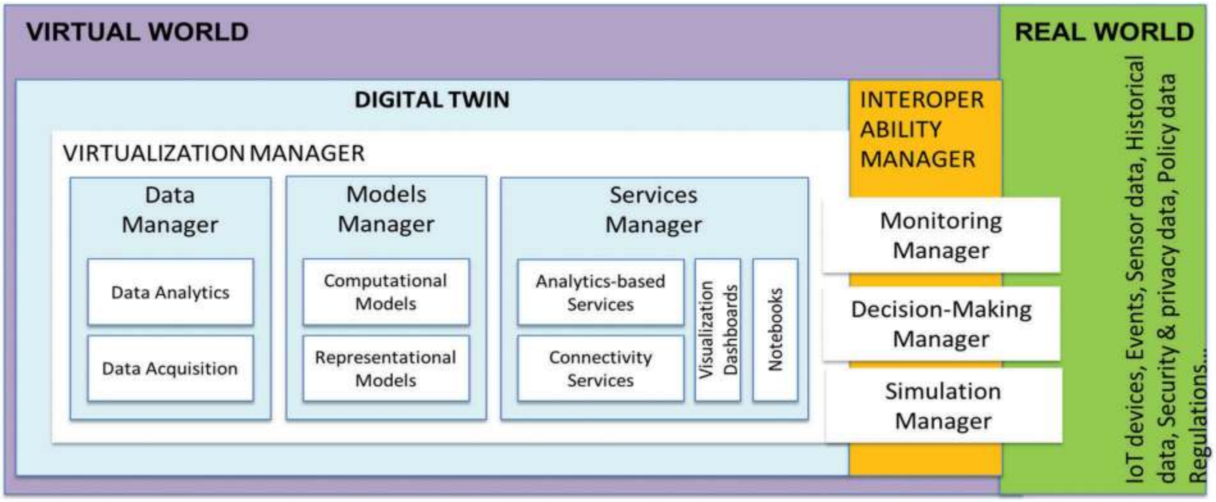

Figure 1. Conceptual model of the DT demonstrator.

- Data analytics: The collected data of DTs can be either structured, unstructured or semi-structured. The data can be ingested as a stream in real-time, or as batch-oriented data generated from various sources. The data are often heterogeneous. In DTs, the bigger the diversity of the collected data that the ML model has to analyse and learn about the states that matter along the manufacturing path, the better the model will be. For example, the availability of historical data helps ML models to learn the maintenance states of assets for predictive maintenance. In addition, the collected data could be used to predict product and process related behaviour, optimise manufacturing processes, discover anomalies, perform MRO (maintenance, repair, and overhaul) processes, etc.

The process of storing and preparing data for processing requires adequate analytics tools to be put in place, e.g. data can come to the DT system from Hadoop data clusters, SQL data exports, Kafka messaging server or other data stream processing engines. Data storage components and data formats (e.g. SQL, NoSQL, a data warehouse) can also have a profound impact on the capacity, performance, long-term reliability and durability of the DT data storage infrastructure. Finally, data failover and quality check mechanisms, backup and disaster recovery mechanisms need to be put in place and linked through data management capabilities of the DT.

3.2 Models Manager

The Model Manager of a DT includes data representation models (static, structural models) and data computation models (dynamic, behaviour models) (see Figure 1).

- Data representation models are used for storing, exchanging and searching data. They include (Schroeder et al. 2016): (i) semantic data models, e.g. ontologies and taxonomies for sharing PLM knowledge (e.g. Young et al. 2007); (ii) Extensible Markup Language (XML)-based models for encoding documents in a format that is human- and machine-readable (e.g. Choi, Yoon, and Noh 2010); (iii) the STEP model (STandard for the Exchange of Product data) to describe product lifecycle data (e.g. Pratt 2001); (iv) the computer-aided engineering exchange (CAEX), a meta model for the storage and exchange of engineering data models (e.g. Lüder, Hundt, and Keibel 2010). The list can be extended to include some emerging manufacturing data representational models (Von Euler-Chelpin 2008), e. g. PLM XML, an open format from Siemens for facilitating PLM; ASME B5.59-2 standard that addresses performance and capabilities of machine tools at any time in their lifecycle phases, e.g. during specification, after acceptance testing, or during operation; ISO 16739 (IFC) defines a common data model for building lifecycle support that can be applied to manufacturing facilities; IEC 62890 defines standards for lifecycle management for systems and products used in industrial process measurement, control and automation and many more.
- Data computation models perform analytics and processing along the product lifecycle phases, supporting, e.g. system models, functional models, 3D geometric models, manufacturing computation models, usage models (Schroeder et al. 2016; Rios et al. 2015). In DTs, the data computation models need to support continuous learning and improvement based on run-time data gathered from the operating CPSs. Practically, the collected data should be used to further improve simulation quality and adapt DTs to contextual changes occurring in the system. By employing model learning algorithms (e.g. Deep Learning for Neural Networks to learn anomalies of the system), the inferred data learned during the run time can be incorporated into the DT knowledge base and continuous learning and improvement features of the DT can be further experimented to support a range of stakeholders involved in planning and designing, modifying, optimising and verifying industrial factory settings and processes.

3.3 Services Manager

The Services Manager of a DT requires a scalable and modular infrastructure to enable intelligent composition and orchestration of services. DT services may vary significantly, depending on business models and use cases, desired system capabilities, the role of stakeholders interacting with the DT. For example, different stakeholders like operators, engineers, manufacturers, suppliers, customers, maintainers could all be interested in exploiting the DT, but each one from a different perspective.

Examples of DT services related to Smart Manufacturing domains are:

- production services for real-time state monitoring of the physical product, its environment and processes; real-time data management and asset management; real-time user management and user operations; real-time product failure analysis and prediction (anomaly detection); real-time behaviour analysis that can help manufacturers to improve product and production performances, e.g. condition monitoring, real-time image processing, etc.;
- supply chain control services that need to serve multiple tenants simultaneously; services that predict supply chain performances, etc.;
- cybersecurity services application security application (authentication, authorisation, etc.), maintaining awareness of the security and privacy conditions through continuous monitoring processes of a DT and more.

4. Open source components for the Digital Twin demonstrator

In the last 30 years, open source technology has become very popular and economically important in the software industry. According to recent statistics on open source tools provided by DZone (2018), open source initiatives prove their helpfulness (with 74% of respondents claiming that), enjoy popularity with developers in general (with 68%) and are often used because of the maturity of their solution (with 62%). The most important reasons for developers to choose open source tools relate to welcoming communities (for 54%), reduced development costs (for 80%) and no vendor lock-in solutions (for 59%). Open source projects generate high-quality code, enable freedom to adapt and collaboratively improve code, collaboratively inspect code for security issues, discover and fix vulnerabilities. When it comes to collaborative development and free distribution of code, the risks of creating vulnerabilities in the system are constantly present. Hence, testing for vulnerabilities in open source components at every release has become one of the best security practises to keep the system safe and ensure that the code is secure and will keep any operational data protected. The complexity of open source tools and mixed licensing obligations (mostly arising from permissive vs. viral open source offerings) are some common problems with open source tools (DZone 2018)). It is therefore advisable to either keep to permissive licences or to be well aware of how and when to use non-permissive ('viral') open source components.

Our description of open source tools follows the conceptual view presented in the previous section, i.e. addressing data management (data acquisition, data exchange, data streaming); data representation models (e.g. semantic models or ontologies) and services for analytics, operational optimisation methods, etc.

4.1. Open source tools for data management of Digital Twins

Data acquisition systems (DAS) are one of the crucial elements for the implementation of a DT for manufacturing environments (Uhlemann et al. 2017)). The data can be collected through measurements and database queries (e.g. non-volatile data capturing a specification of equipment on the factory floor, lists of products and bills of materials, etc.) and through real-time sensor-based processing systems (e.g. volatile data capturing movement of objects on the factory floor, human motion, flow of material, processing time and capacity of machinery, etc.).

In DTs, an efficient DAS requires high levels of connectivity in factories, as a prerequisite to enable data exchange. The most used

Table 1. Open source tools for data exchange and connectivity in Smart Manufacturing.

Name	Description	Link
MQTT (Message Queue Telemetry Transport)	Machine-to-machine (M2M)/IoT connectivity protocol and a publish/subscribe messaging transport protocol that is optimised to connect physical world devices and events with various enterprise servers.	https://github.com/mqtt/mqtt.github.io
AMQP (Advanced Message Queuing Protocol)	An OASIS open standard for passing business messages between applications and organisations. It provides a publish/subscribe architecture that is similar to MQTT.	https://www.amqp.org/
CoAP (Constrained Application Protocol)	A specialised web transfer protocol for constrained node devices and networks in the IoT.	http://coap.technology/
OPC UA (OPC Unified Architecture)	A data exchange standard that offers a secure data exchange and allows platform-independent and vendor-neutral communication (OPC Foundation 2016).	http://open62541.org
OneM2M	A global standard created to reduce the diversity of devices and legacy systems.	http://www.onem2m.org/ oneM2M (2018)
Eclipse Mosquitto	An open source (EPL/EDL licenced) message broker that implements the MQTT protocol versions 3.1 and 3.1.1. Mosquitto is suitable for use on all devices from low power single board computers to full servers.	https://mosquitto.org/
XMPP (Extensible Messaging and Present Protocol)	An open XML technology for real-time communication.	https://xmpp.org/
OpenDDS	An open source implementation of the OMG's DDS (Data Distribution Service) for real-time systems	http://opendds.org/
OMA LWM2M (Lightweight M2M)	An open source implementation for sensor networks and M2M (Machine to Machine) communication.	https://www.omaspecworks.org/

Table 2. Open source middleware platforms for software deployment.

Name	Description	Link
Docker	A software container management platform.	https://docker.com/
Kubernetes	An open source platform for automating deployment, scaling and management of containerised applications.	https://kubernetes.io/
Eclipse Kapua	A platform for management and integration of devices and data.	https://www.eclipse.org/kapua/
Eclipse Kura	A connectivity middleware.	http://www.eclipse.org/kura/

open source tools for data exchange and connectivity for Smart Manufacturing are summarised in Table 1.

Application programme interface (API) technology and data streaming platforms are often used to connect disparate software components. Table 2 presents the most popular open source middleware platforms for software deployment.

Data streaming systems enable streaming processing and perform actions on real-time data through the use of continuous queries (Freeman 2016). The most important open source streaming processing tools are summarised in Table 3.

4.2. Data representation models for Digital Twins

In DTs, domain knowledge about the factory floor processes and equipment needs to be modelled and integrated into manufacturing applications. Here, the ontologies are seen as natural candidates for implementing knowledge-based systems (KBS), which formalise knowledge about a domain (Giovannini et al. 2012)). The role of ontologies is to capture a formal and shared representation of a particular domain of disclosure, which can be used in a variety of Smart Manufacturing fields.

Some of the best-known ontologies in manufacturing domains are summarised below:

- Cai, Zhang, and Zhang (2001) present an ontology-based solution to demonstrate the interoperability between manufacturing services.
- There have been several ontologies developed to formally model manufacturing systems. For example, Diep, Alexakos, and Wagner (2007) present the P2 Ontology, whose aim it is to allow interoperability between components and applications throughout the manufacturing process life cycle; Lemaignan et al. (2006) created MASON, an upper ontology of manufacturing systems; the Process Specification Language (PSL) Ontology was designed to facilitate the exchange of process information among manufacturing systems and has been published as ISO 18629 by the International Organisation of Standardisation (Menzel and Gruninger (2001), Schlenoff et al. (2000)).
- Chang, Rai, and Terpenny (2010) present the design for manufacturing (DFM) Ontology that captures relevant domain manufacturing knowledge, enhances the knowledge exchange and retrieval of manufacturing design alternatives from heterogenous data sources and supports designers in making design decisions.
- To achieve fast and flexible reconfiguration of modular manufacturing systems, Alsafi and Vyatkin (2010) present a reconfiguration agent that is based on the MASON Ontology and designed to infer knowledge about the manufacturing environment and its requirements.
- Ameri, Urbanovsky, and McArthur (2012) propose a systematic approach for the development of manufacturing ontologies based on MSDL (Manufacturing Service Description Language) and the MSDL Ontology, which enhances formal representation of manufacturing services in mechanical machining domain.
- Kiritsis et al. (2013) designed the LinkedDesign Ontology that can be adjusted and adopted for different manufacturing systems.
- Chungoora et al. (2013) explore the core ontological concepts encoded in the ECLIF (Extended Common Logic Interchange Format) (ECLIF (2010)) format for application configurations of products and information platforms in manufacturing domains.
- The P-PSO (Politecnico di Milano—Production Systems Ontology) provides a meta-model of various manufacturing system domains and applications (Garetti and Fumagalli (2012), Garetti and Fumagalli (2012a), Garetti, Fumagalli, and Negri (2015)). The P-PSO Ontology has evolved into the MSO (Manufacturing Systems Ontology) for logistics, discrete and production manufacturing systems and processes (Negri et al. 2015a; Negri et al. 2017).

Table 3. Open source streaming processing tools.

Name	Description	Link
Apache Spark Streaming	An extension of the core Apache Spark API for stream processing of live data streams.	https://spark.apache.org/streaming/
Apache Storm	An open source real-time distributed processing platform developed by Twitter, built without batch processing capabilities.	http://storm.apache.org/
Apache Kafka Streams	A lightweight library used for streaming data coming from Apache Kafka.	https://kafka.apache.org/
Apache Samza	A full-fledged cluster processing software. It runs on Apache Hadoop YARN (Yet Another Resource Negotiator).	http://samza.apache.org/ http://hadoop.apache.org/
Apache Flink	A framework for unified stream and batch processing.	https://flink.apache.org/
Apache Beam	A unified programming model for stream and batch use cases.	https://beam.apache.org

Some of the ontology-based manufacturing applications have been recently presented in the literature:

- Garetti et al. (2013) discuss the integration of ontologies and Web Services within the control architecture of automated manufacturing systems.
- Negri et al. (2015) present the core requirements for the use of manufacturing domain ontologies in a Web Service architecture for the control of manufacturing systems.
- To design and implement a flexible architecture for event-driven manufacturing systems that can be deployed in multiple industrial cases, Mohammed et al. (2017) combine the flexibility of knowledge-driven systems with the vendor-independent properties of RESTful Web Services.

- Zhang et al. (2012) present the Cloud Recommender System based on an OWL (Web Ontology Language) Ontology.
- Afify et al. (2013) discuss a Software-as-a-System discovery and selection system based on the WordNet ontology.
- Ontology-based service matching in Cloud computing is also discussed in Liu et al. (2014).
- Rodríguez-García et al. (2014) present an ontology-based annotation and service retrieval system for the Cloud environment.

Semantic integration of sensor data has been explored through many efforts to create sensor taxonomies, ontologies and standards. Table 4 shows some of the most prominent open data formats and ontologies of relevance for Smart CPSs and DTs.

4.3. Open source tools and libraries for computational models of Digital Twins

Computational models are designed either for batch processing or for real-time data processing. The most important open source tools for batch-oriented processing are briefly presented in Table 5.

Oussous et al. (2017) provide a comprehensive comparison between HDFS and HBase features.

Time-series processing is often used for industrial monitoring and processing or for tracking corporate business metrics (NIST Statistics 2012). One of the first challenges to solve when designing a temporal data-generating system is to decide on the right storage engine to be used for time-series data. Another challenge is about methods and tools for querying and aggregating a large amount of sensor data to extract useful information. Some popular open source tools for the computation of the advanced time-series data are given in Table 6.

Table 4. Open data formats and ontologies for Smart Manufacturing.

Name	Link and references
W3C SSNO (Semantic Sensor Network Ontology) created by the W3C Semantic Sensor Network Incubator group (SSN-XG)	https://www.w3.org/TR/vocab-ssn/
Ontology for adaptive sensor networks	Avancha, Patel, and Joshi (2004)
SensorML (Sensor Modelling Language) created under the Open Geospatial Consortium (OGC) Sensor Web Enablement (SWE) activity	http://opengeospatial.org/standards/sensorml
OntoSensor ontology with extensions to IEEE SUMO (Suggested Upper Merged Ontology)	Russomanno, Kothari, and Thomas (2005)
CESN (Coastal Environment Sensor Network) ontology	http://www.cesn.umb.edu/sensor/cesn.owl
CSIRO sensor ontology	Neuhaus and Compton (2009)
W3C SSNO (Semantic Sensor Network Ontology)	https://www.w3.org/TR/vocab-ssn/
W3C SOSA (Sensor, Observation, Sample, and Actuator)	http://www.w3.org/ns/sosa/
AutomationML, an open, XML-based and standardised data format	https://automationml.org
SensorThings API, an international standard for connecting IoT devices over the Web	http://developers.sensorup.com/docs/
onem2m Base Ontology for semantic interoperability between oneM2M-based systems and external systems	https://git.onem2m.org/MAS/BaseOntology oneM2M (2018)

Table 5. Open source tools for batch-oriented processing.

Name	Description	Link and references
Apache Spark	An in-memory distributed data processing platform for large-scale data processing and batch analysis jobs.	https://spark.apache.org/
Apache Hadoop	A high-throughput system that can process large volumes of data using a distributed parallel processing paradigm, called MapReduce (for batch queries).	https://mapr.com/
HDFS (Hadoop Distributed File System)	A data storage system that provides cost effective and reliable storage capability. It can handle both structured and unstructured data.	http://hadoop.apache.org/hdfs/ White (2012)
Apache HBase	A distributed non-relational database. It is built on top of HDFS and designed for low latency operations.	Prasad and Agarwal (2016)

Table 6. Open source tools for computation of advanced time-series data.

Name	Description	Link and references
R project	An open source project that is widely used by many researchers and data scientists	https://www.r-project.org/ (McLeod, Yu, and Mahdi 2011)
InfluxDB	A database for storing time series data. It supports data transformation and prediction queries. The tools like Grafana can be used to configure access to InfluxDB data and support visualisation.	https://www.influxdata.com/
Elastic Stack (ELK Stack)	It offers many tools for searching, analysing and visualising data in real-time. It comprises of Elasticsearch, Logstash, Beats and Kibana.	https://www.elastic.co/elk-stack
Elasticsearch	It handles textual data, which is helpful for analysis of logs that contain messages, exceptions, and other text-based information.	https://www.elastic.com Berman 2017
Logstash	An open source processing pipeline that is used to import logs from different sources, transform and output them to a defined endpoint or data store.	https://www.elastic.co/products/logstash
Kibana	The ELK Stack's User Interface that provides a visual representation of the stored log files and offers customisation of dashboards, tables, etc.	https://www.elastic.co/products/kibana
Beats	Beats are open source data shippers used for capturing various data (e.g. network traffic, metrics, cloud data, etc.) and sending them to Elasticsearch.	https://www.elastic.co/products/beats

Table 7 summarises some of the most widely used open source tools for ML and analytics, including statistical packages for ML, optimisation, etc.

5. Architecture design of a Digital Twin demonstrator

The proposed design of the DT demonstrator follows the micro-services architectural paradigm at the implementation level, and the IIRA and the RAMI 4.0, at the conceptual level. Firstly, we present the motivation for using a micro-services architecture. Secondly, the mapping between the proposed conceptual model defined in Section 3, and IIRA and RAMI 4.0

Table 7. Open source tools for ML and analytics.

Name	Description	Link and references
Apache Mahout	A framework for creating ML algorithms and applications.	http://mahout.apache.org/
Apache OpenNLP	An open source Java library for processing natural language text.	http://opennlp.apache.org/
Apache Spark MLlib	An open source library with ML algorithms	https://spark.apache.org/mllib/
Apache Spark GraphX	An open source library for graph-based computation	https://spark.apache.org/graphx/
SciPy	A collection of open source software for scientific computing in Python	https://scipy.org/
R project	Statistical computing and graphics in R language	https://www.r-project.org/
TensorFlow	An open source library for designing, building and training Deep Learning models and high-performance numerical computation and ML.	https://github.com/tensorflow/tensorflow https://www.r-project.org/
ThingsBoard	A platform for data collection, processing, visualisation, device management, asset management, customisation	https://thingsboard.io/
ML and Deep Learning frameworks	Keras Caffe Kaggle PyTorch Torch/Lua	http://keras.io/ http://caffe.berkeleyvision.org/ https://www.kaggle.com/ https://github.com/pytorch http://torch.ch/
Notebooks for customised data analytics	Apache Zeppelin for analytics in SQL, Anaconda Jupyter CoCalc for mathematical calculation, statistics and data science ().	https://zeppelin.apache.org/ https://anaconda.org/anaconda/jupyter https://cocalc.com/

is provided. Thirdly, the specific micro-services architecture for DTs is presented.

5.1. Motivation for implementing Digital Twins as micro-services

In contrast to monolithic systems, which are often built as a massive code base, the micro-service architectural style enables a single application to be developed as a suite of relatively small, consistent, isolated and autonomous services, each performing a specific task (Lewis and Fowler 2014). Micro-services can be developed and deployed independently by different teams, and are language agnostic. Taibi, Lenarduzzi, and Pahl (2017) provide an analysis of the survey on major motivations for migrating from monoliths to micro-services. For example, in the survey, software maintenance has been reported as very important by all the participants. Scalability of micro-services, delegation of team responsibilities to other teams and the easy support for DevOps were also highly rated.

In the case of Smart CPSs and DTs, which are complex, non-linear systems that require a variety of mechanisms to represent their static artefacts and dynamic capabilities, the adoption of the modular architecture of micro-services allows the application complexity to be reduced and code to be better maintained. However, the research in DT has raised a number of new challenges, e.g. developing computationally efficient algorithms for predicting system behaviour in real-time, edge data processing tools, dealing with uncertainty in the system, etc. Micro-services as a design choice offer flexibility and offer potential to reduce the complexity of DT systems.

At the same time, using micro-services can possibly open security vulnerabilities and threaten the trustworthiness of services, and this requires a good design balance between security and system performance (Esposito, Castiglione, and Choo 2016).

5.2. Mapping of the Digital Twin conceptual architecture to Smart Manufacturing reference models

As Smart Manufacturing systems are becoming increasingly interconnected and complex, CPS and DTs must be architected, designed and implemented to enable integration of heterogeneous technologies and their effective interoperation. Lin et al. (2017) discuss how the two reference architectures (models), IIRA and RAMI 4.0 relate to one another. They show that the two models complement each other more than they are conflicting, and that they can be mapped to each other despite each being based on different architecture framework standards. The general understanding is that IIRA emphasises applicability and interoperability across industries, while the RAMI 4.0 is more focused on digitalisation of Smart Manufacturing. In this section, we provide an alignment of IIRA and RAMI 4.0, with the proposed DT conceptual architecture from Section 3. The architectural alignment takes IIRA's cross-cutting functions and RAMI's layers as reference for the entire DT conceptual architecture as illustrated in Figure 2.

RAMI 4.0 has the following six layers: asset, integration, communication, information, functional and business layer (ZVEI 2015). The IIRA cross-cutting functions are: connectivity, distributed data management, industrial analytics, intelligent and resilient control (Industrial Internet Consortium 2017). The mapping of the DT conceptual model with RAMI 4.0 and IIRA is based on the following facts:

- The asset layer in RAMI 4.0 refers to anything that participates in the business processes (sensors, machines, raw material, software, human actors). The DT conceptual architecture refers to the real-world assets, e.g. sensor data, IoT devices, stakeholders, etc., which are monitored through the Monitoring Manager of the DT Virtualisation Manager. Hence, the asset layer in RAMI 4.0 can be directly correlated to the DT real-world assets.
- Both the communication and integration layers of RAMI 4.0 provide communication standards for services, event/data and control commands that link the physical assets and their digital capability. In IIRA, the connectivity function points to standards like DDS, OneM2M, etc. In the DT conceptual architecture, enabling communication and integration of various subsystems is a task of the Interoperability Manager.
- The information layer in RAMI 4.0 describes the services and data that are offered, used, generated or modified through the asset. In IIRA, the distributed data

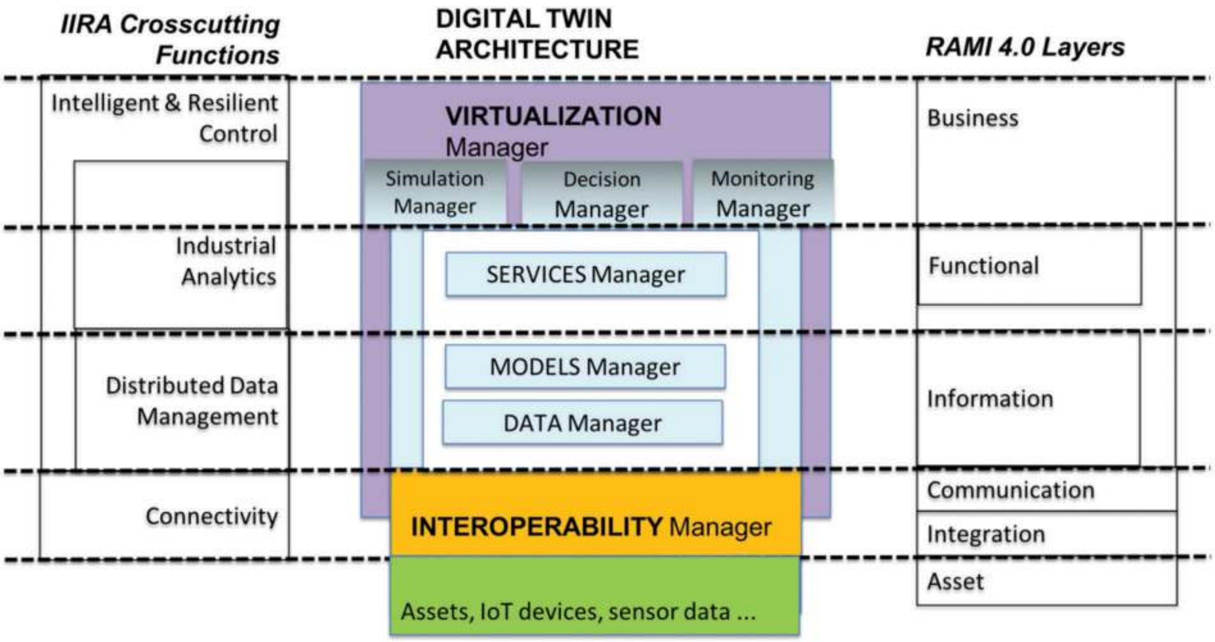

Figure 2. Architectural alignment between the IIRA reference model, the DT conceptual model and the RAMI 4.0.

management function is about data management, which corresponds to the Data Manager and the Models Manager of the DT concept architecture.

- The functional layer in RAMI 4.0 describes logical functions of an asset that differ according to their role in Smart Manufacturing. This corresponds to IIRA's industrial analytics function, and the Services Manager of the DT concept architecture.
- The business layer in RAMI 4.0 creates business processes and orchestrates these processes to enable business models under specific legal and regulatory constraints. In IIRA, the intelligent and resilient control function enables intelligent controls, which is the main focus of the DT' Virtualisation Manager and its components for simulation, decision-making and monitoring (c.f. Simulation Manager, Decision-Making Manager and Monitoring Manager).

The proposed DT conceptual architecture bears close relationship to both IIRA and RAMI 4.0 as shown in Figure 2.

5.3. Micro-services architecture of the Digital Twin demonstrator

The ability of Smart CPSs to interconnect and merge into the ubiquitous Web of Things (WoT) infrastructure (e.g. Cloud computing and cloud services, smart gateways and network edge devices) is one of the critical requirements of Smart Manufacturing. While the WoT facilitates reuse of current web technologies for future application development, the challenge is still to create an effective infrastructure to process data and provide service and application flexibility, e.g. to maintain 'degrees of freedom' for new services while the smartness of systems increases their complexity.

The approach presented in this paper follows the micro-services architectural style, in order to enable decomposition of the service- and application-logic and reduce the complexity of DTs into smaller partitions of flexible, functionally independent and executable services. The proposed architecture of the DT demonstrator follows the conceptual model shown in Figure 1. It consists of the following building blocks (Damjanovic-Behrendt 2018): Virtualisation Manager, Data Manager, Models Manager, Services Manager and Interoperability Component (see Figure 3), each of them encompassing a set of defined micro-services.

The rest of this section describes each of the DT micro-services building blocks.

The **Virtualisation Manager** is composed of the micro-services described in Table 8. For example, it enables monitoring of factory floor assets and events through its Monitoring Manager. Through decision support services and controls, it detects conflicts and automatically enables their resolution.

The **Data Management Component** is composed of micro-services dealing with data acquisition, data analytics and knowledge discovery (Table 9). Knowledge Discovery micro-services require various analytics methods to be put in place. The results of analytics methods need to be further exposed to simulation and visualisation micro-services.

The **Models Manager Component** includes services for the definition, execution and maintenance of data computation and data representation models (Table 10). Services for data computation are further coupled with data analytics services. Services for data representation are maintained either as semantic models (e.g. described in resource description framework (RDF)) or relational models (based on relational databases).

The **Services Management Component** releases IoT/WoT connectivity services, offers services through notebooks for customised analytics and performs cybersecurity tasks that target data access and usage controls, threat detection service, threat analysis service, incident sharing and incident response service (Table 11).

The **Interoperability Component** of the DT demonstrator is designed to offer interoperability mechanisms at the data

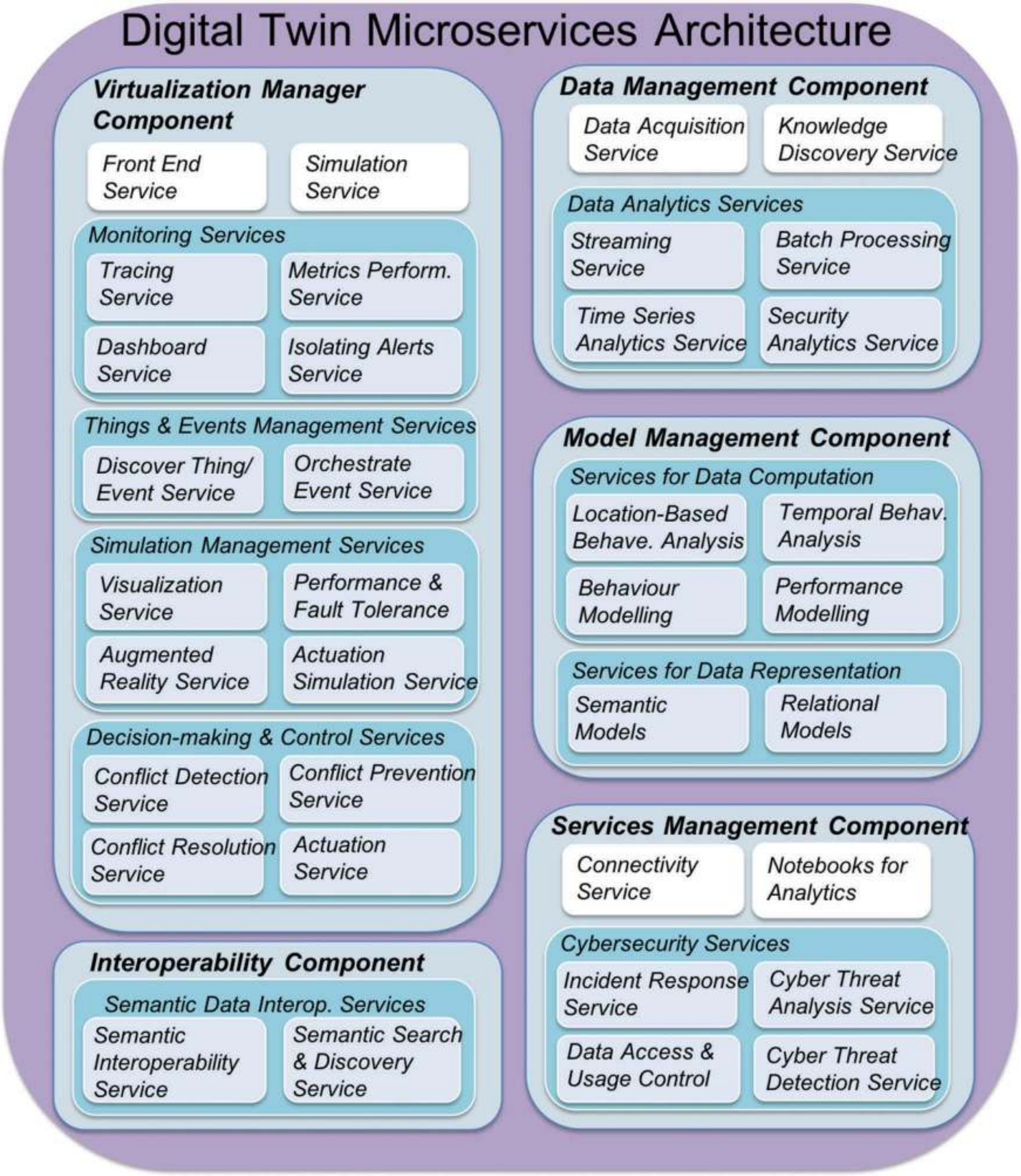
Digital Twin Microservices Architecture
Virtualization Manager Component
Front End Service
Simulation Service
Monitoring Services
Tracing Service
Metrics Perform. Service
Dashboard Service
Isolating Alerts Service
Things & Events Management Services
Discover Thing/ Event Service
Orchestrate Event Service
Simulation Management Services
Visualization Service
Performance & Fault Tolerance
Augmented Reality Service
Actuation Simulation Service
Decision-making & Control Services
Conflict Detection Service
Conflict Prevention Service
Conflict Resolution Service
Actuation Service
Interoperability Component
Semantic Data Interop. Services
Semantic Interoperability Service
Semantic Search & Discovery Service
Data Management Component
Data Acquisition Service
Knowledge Discovery Service
Data Analytics Services
Streaming Service
Batch Processing Service
Time Series Analytics Service
Security Analytics Service
Model Management Component
Services for Data Computation
Location-Based Behave. Analysis
Temporal Behav. Analysis
Behaviour Modelling
Performance Modelling
Services for Data Representation
Semantic Models
Relational Models
Services Management Component
Connectivity Service
Notebooks for Analytics
Cybersecurity Services
Incident Response Service
Cyber Threat Analysis Service
Data Access & Usage Control
Cyber Threat Detection Service

Figure 3. Micro-services architecture of the DT demonstrator.

level. In DTs, interoperability services are critically important for enabling the usage of the factory floor devices (and their data), produced by different manufacturers. Interoperability services ensure implementation of DTs, and their simulation functionalities.

McCool (2017) addresses the following three levels of interoperability for WoT:

- semantic interoperability (decoding the meaning of data),
- structural interoperability (decoding the organisation of data), and
- syntactic interoperability (converting data in a consistent way between a serialised representation and an internal data structure (e.g. a parse tree)).

To enhance usability of data and data models in DTs, the proposed DT demonstrator put its emphasis on semantic data interoperability, which is supported through the semantic interoperability service, and the semantic search and discovery service (Table 12).

6. Case study infrastructure for the implementation of the Digital Twin demonstrator

Knowledge of common and distinct features of open source software is an important factor when deciding which of the available tools to choose. For example, the process of capturing, storing and managing a large amount of time-series data requires considerable effort and research on combining open

Table 8. Micro-services of the Virtualisation Manager Component.

Name	Description
Front-End Service	Establishes the front-end Web components of micro-services. These are reusable components that can be imported into Web apps. The guidelines for creating web platform compatible Web components are given in W3C TAG (2018).
Monitoring Services	A set of micro-services that helps developers to understand the system behaviour by breaking the system down into smaller applications, e.g. Tracing Service, Metrics Performances Service, Isolating Alerts, Dashboards Service.
• *Tracing Service*	Supports continuous tracking and tracing of factory shop floor assets across various subsystems, e.g. supply chains, ERP, Manufacturing Execution Systems (MESs), etc.
• *Metrics Performances Service*	Measures performance and use of Operational Technology (OT) assets on the manufacturing shop floor.
• *Isolating Alerts Service*	Provides controls for detecting and isolating problems that target specific processes/assets. It is coupled with Cybersecurity Services (of the Service Management Component) to create cross-layer alerts based on cybersecurity analytics.
• *Dashboards Service*	Enables visualisation dashboards for monitoring interactions, continuous integration, cross-service visibility, etc.
Things and Events Management Services	Enables discovery of things/assets and events, and the orchestration of events on the manufacturing shop floor.
• *Discover Things/Event Service*	Allows for device functionality to be dynamically discovered and optimally exploited. It also supports the running events on the manufacturing shop floor to be discovered, for further services and decisions.
• *Orchestrate Event Service*	Allows for creating more cooperative manufacturing models through effective orchestration.
Simulation Management Services	A set of micro-services that incorporate performance measures and observations received from the physical world in order to manage simulation inputs for the DT.
• *Visualisation Service*	Enables visualisation of measured performances of the system, e.g. through dashboards.
• *Augmented Reality Service*	Enables simulations using Augmented Reality technologies.
• *Actuation Simulation Service*	Enables simulations of a specific actuation.
• *Performance and Fault Tolerance Service*	Measures performances and fault tolerance of the manufacturing system.
Simulation Services	Erforms the simulation based on the formats defined by the *Simulation Management Services.*
Decision-Making and Control Services	A set of micro-services for specific decision support and controlling functionalities of DTs.
• Conflict Detection Service	Enables the identification of assets and events from the manufacturing shop floor, which significantly differ from the majority of relevant data (based on insights and measurements). Conflicts can be referred to as noise, or deviations.
• Conflict Resolution Service	Based on identified conflicts, their nature, durability and other detected features, this service is in charge of providing the adequate resolution strategy.
• Conflict Prevention Service	Through monitoring and analytics, this service provides mechanisms to avoid conflicts in the manufacturing system, caused by noise and deviation.
• Actuation Service	Ensures that feedback created by DT mechanisms is transmitted to the real manufacturing environment.

Table 9. Micro-services of the Data Management Component.

Name	Description
Data Acquisition Service	Enables data acquisition for DTs, e.g. data collected through sensors and from tracking and tracing technologies needs to be stored and maintained for warranty and other purposes.
Data Analytics Services	Enables various data analytics services, e.g. ML-based analytics for predicting assets' behaviour within the changed manufacturing environment.
• *Streaming Service*	Supports streaming process analysis.
• *Batch Processing Service*	Supports batch-oriented processing.
• *Time Series Analytics Service*	Supports time series-based analysis.
• *Security Analytics Service*	Supports security analytics and is further coupled with Cybersecurity Services of the Service Management Component.
Knowledge Discovery Service	The analytics techniques provide feedback mechanisms that send decisions and responsive actions back to the DT and physical system.

Table 10. Micro-services of the Models Manager Component.

Name	Description
Services for Data Computation	These services enable the major analytics processes of DTs.
• *Location-Based Behavioural Analysis*	Typical services for location-based behavioural analysis allow for the factory floor assets to be identified based on their spatial location. The location of assets can be shared with other assets and events in the manufacturing ecosystem. These services include location prediction, location-based asset management, recommender systems, etc.
• *Temporal Behaviour Analysis Service*	Ensures temporal localisation of the factory floor assets and events. Some popular methods include simulation techniques and discrete event systems (e.g. Petri nets).
• *Performance Modelling Service*	Enables modelling of performances of the production line processes and assets.
• *Behavioural Modelling Service*	Enables modelling of behaviour of the production line processes and assets under specific conditions of the manufacturing environment.
Services for Data Representation	These services support the inclusion of various data representation formats in DTs
• *Services for Semantic Models Management*	Inclusion of the relevant manufacturing ontologies in the knowledge base, semantic services, semantic reasoning, ontology management for DTs, etc.
• *Services for Relational Models Management*	Support of the management of relational data models; data interfaces and integration mechanisms for heterogeneous databases, etc.

source technologies for the storage mechanisms and the data analytics engine. For example, Churilo (2018) discusses the results of recent benchmark for time-series workloads for InfluxDB 1.4.2, an open source time-series database and Elasticsearch 5.6.3 (see Table 7 in Section 4.3 for details). The performed benchmark shows that InfluxDB outperforms Elasticsearch in two tests:

- write throughput (InfluxDB is 9.9x greater than Elasticsearch) and
- disc space usage (InfluxDB uses 13.1x less disc space when compared against Elasticsearch's time-series optimised configuration).

Table 11. Micro-services of the Services Management Component.

Name	Description
Connectivity Service	Ensures connectivity with factory floor assets, products and stakeholders involved through manufacturing life cycle processes. A comprehensive and systematic review of the literature is presented in Liao et al. (2017), the most widely used standards and protocols for connectivity between machines in Smart Manufacturing are OPC, MQTT, IEC 62439, IEEE 1588, IEC 61850.
Notebooks for Analytics	Services that enable creating of notebook files that contain Python, Scala, R code and markdown text, which can be shared with other analytics services of DTs.
Cybersecurity Services	Services that enable cybersecurity analytics and management, including incident response, threat analysis, access controls and governance models.
• *Incident Response Service*	Enables different levels of security for the DT based on the sensitivity of data it carries. Smart CPSs are also vulnerable to security threats and privacy breaches that stem from (open source) communication technologies and protocols, and hence they require the right strategy to be followed to improve cybersecurity. Some examples of open source tools in this category are: • STIX (Structured Threat Information eXpression) language and serialisation format for Cyber Threat Intelligence (CTI) (https://github.com/STIXProject/) • MISP (Malware Information Sharing Platform) for sharing CTI (https://github.com/MISP) • MAEC (Malware Attribute Enumeration and Characterisation) language for sharing information about malware (https://github.com/MAECProject/).
• *Cyber Threat Analysis Service*	Cybersecurity threat analysis for a DT is based either on fully automated tools designed to quickly assess network traffic, file activity, etc. and contribute to the incident response process, or on static analysis of malware properties. An examples of open source tools in this category would be the Cuckoo Sandbox for automated malware analysis (https://cuckoosandbox.org/).
• *Data Access and Usage Control*	In DTs, cloud service providers, users and often fog devices as tenants, do not trust each other. Hence, access and usage controls for data and services in DTs require well defined access control policies to preserve user privacy and ensure system security. Services in this category need to support Virtual Machines (VMs) e.g. providing an access control mechanism to avoid side-channel attacks, and to provide access controls for the fog and cloud, reciprocally (Zhang et al. 2018).

Table 12. Micro-services of the Interoperability Component.

Name	Description
• *Semantic Data Interoperability Service*	This service is based on the W3C Web of Things (https://www.w3.org/WoT/) set of semantics and metadata standards around IoT. The focus of this service is on converting WoT representations that include identifiers, properties, and relationships into the meaning of data, through shared contexts, vocabularies, and ontologies (iot.schema.org, SSNO, SAREF, and many more.)
• *Semantic Search and Discovery Service*	In DTs, apart from ML-based processing of data, semantic search and reasoning capabilities are beneficial, too. This service ensures (i) rules and semantic alignments to transform data to the declared ontologies, e.g. using JSON-LD for RDF data serialisation, and (ii) reasoning engines for inferring associations and links into the data (Szilagyi and Wira 2016).

On the other hand, InfluxDB delivered 20% slower response times for tested queries, compared to response time of cached queries from Elasticsearch (Churilo 2018). In the case of querying stored log messages, requests, responses and exceptions in the system, Elasticsearch is a better solution than InfluxDB, because storing and querying log data on InfluxDB requires adding another search engine. Therefore, creating an architecture that includes both InfluxDB and Elasticsearch operating in parallel, is an option for the implementation of systems based on time series, that require log data analyses to be performed too.

In situations when large amounts of incoming data are expected, Apache Kafka is often used as an event store engine that maintains ordered sequences of entries, allowing multiple consumers to pull in the data and process it. Dobbelaere and Esmaili (2017) perform a qualitative and quantitative comparison of the common features of Apache Kafka vs. RabbitMQ, two popular open-source and commercially supported pub/sub messaging systems. The best suited use cases for Apache Kafka are those implementing pub/sub-messaging with simple routing logic; scalable ingestion systems enabling high throughput processing of stored data; capturing change feeds and stream processing (with Kafka Streams). RabbitMQ is often used as pub/sub-messaging with complex routing logic, or for operational metrics tracking for real-time processing.

Use cases that combine Apache Kafka and RabbitMQ can be implemented in two forms:

- RabbitMQ followed by Apache Kafka (offers stronger latency guarantees) and
- Apache Kafka followed by RabbitMQ (combines the complex routing capabilities of RabbitMQ with the complementary features of Apache Kafka).

The proposed DT demonstrator is designed to be used by Docker and Kubernetes (see Table 2 in Section 4.1 for details) and is configured with the elements as shown in Figure 4.

For example, Apache Kafka (see Table 3 in Section 4.1 for details) allows developers to integrate multiple data sources and systems, e.g. web and mobile applications, APIs and other real-time synchronous and asynchronous systems. The data from databases can be streamed into Kafka via the Kafka Connect API, which requires that data filtering and aggregating are based on KSQL (Kafka SQL) for streaming SQL for Apache Kafka.

RabbitMQ is added as an open source messaging protocol that supports AMQP, MQTT, HTTPS, STOMP and WebSockets. RabbitMQ adds new events in the event stream in real time, which are further sent to Logstash, the dataflow engine in the Elastic Stack that performs data ingestion, enriching and aggregating, regardless of format or schema. Logstash sends data further to Elasticsearch. Data can be sent to Elasticsearch using either its API or ingestion tools such as Logstash, Amazon Kinesis Firehose, Amazon CloudWatch Logs, etc. Elasticsearch stores the original data and adds a searchable reference to it. The data can be further visualised using Kibana, an open-source data visualisation and exploration tool for log and time-series analytics, application monitoring and operational intelligence use cases. Kibana is the default choice for visualising data stored in Elasticsearch.

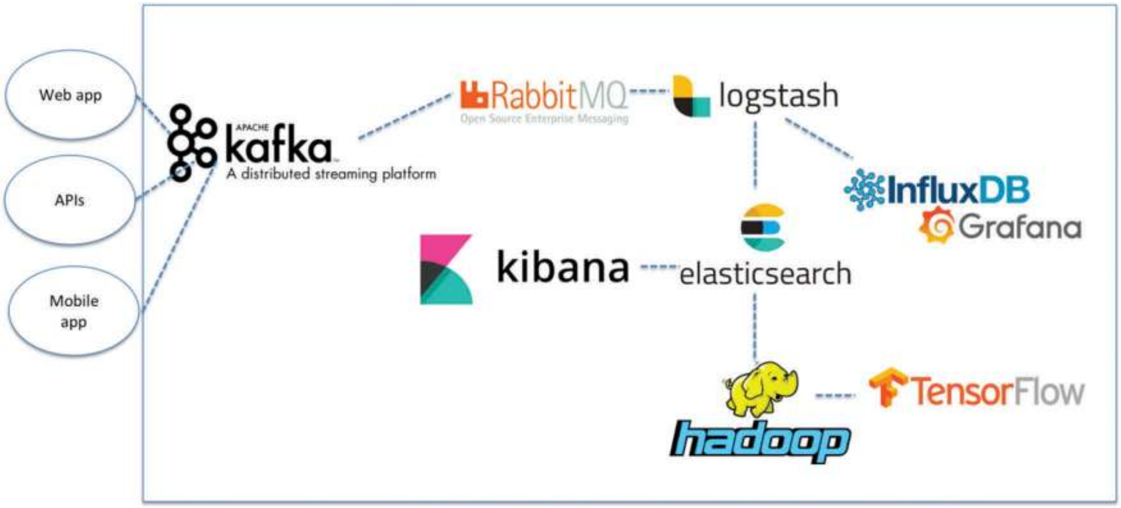

Figure 4. Open source tools composition for the DT demonstrator.

The real-time search and analytics feature of Elasticsearch are further connected with the massive data storage and processing power of Apache Hadoop, and the interoperability between them is currently supported via the Elasticsearch-Hadoop (ES-Hadoop) connector. One of the main advantages of Apache Hadoop is its capacity to rapidly process large datasets, not in memory, but where data are stored (Oussous et al. 2017), which relieves network and servers from a considerable communication load (Usha and A.P.S. 2014). It allows users to add modules as needed, according to their application requirements. Although Apache Hadoop solves problems related to deep and extensive analytics with complex big data, it is not built for real-time processing. Hence, the proposed infrastructure for implementation of the DT demonstrator includes TensorFlow, an open source software library for high performance computation used for ML and Deep Learning. TensorFlow (see Table 7 in Section 4.3 for details) can be used for developing distributed ML models, which can then be trained to offer high performance predictions (e.g. using a Cloud ML Engine).

In addition to Elasticsearch, the proposed infrastructure includes InfluxDB to support time-series workloads (application and performance metrics, network flows and transactional data), which is further connected to Grafana to support visualisation needs. Grafana supports visualisation of numerous metrics for monitoring performance, extracting insights and enabling forecasts.

7. Future research

Although the presented infrastructure is designed for real-world applications in Smart Manufacturing, it is at present, still a research platform that is expected to continuously evolve into a fully operational technology stack in the future. The DT demonstrator needs to be complemented with a comprehensive set of usage methods and validation metrics. In this paper, our work focused on technology building blocks for an open source DT in Smart Manufacturing. As for the set of metrics for the validation of processes and artefacts related to the DT functionality, the work may need to be carried out separately for different sectors within manufacturing. More importantly, the variability of metrics needs to be extended throughout manufacturing lifecycle phases, including the security and privacy lifecycle. Tracking data provenance, handling security and privacy of the data flows through the analysis pipeline and increasing transparency of algorithmic decision making, are important research questions to be addressed in the future.

The role of ontologies in structuring the lifecycle-oriented knowledge base of DTs needs to be further explored, and performance and security metrics need to be integrated into the knowledge base. The availability of large amounts of data in Smart Manufacturing needs to be tested for data quality and requires data governance mechanisms, in order to identify the right amount of data to be processed. For example, some data can be periodically monitored, while others require to be traced in real time. Some data carry critical details for the system functionality, while others contain trivial details. More research is also needed to develop a proper strategy for maintaining the accuracy of DTs, as their effectiveness in both Cloud computing and Fog computing is likely to be compromised. Finally, more research is needed to develop the right synchronisation mechanisms between the virtual space of DTs and the real physical manufacturing assets. The changes within real systems need to be automatically translated to DTs models, and the effective mechanisms to support change management and uncertainty need to be created. Schleich et al. (2017) refer to some additional challenges in DTs, such as currently missing high-fidelity models for simulation and virtual testing at multiple scales, difficulties in the prediction of complex systems, etc.

8. Conclusion

The adoption of CPSs and DTs for automating processes in various domains, ranging from manufacturing to agriculture, is expected to significantly change traditional business models (Serpanos 2018). For example, DT technologies have strong benefits for Smart Manufacturing, enabling monitoring of the execution of simulated lifecycle processes and gaining insights required for informed decisions and predictions, asset management and maintenance. Yet there are still many computational and network challenges to be addressed that relate to the design, operation and management of complex systems based on CPSs and DTs.

The rapid advances in open source technology for data analytics and visualisation have strong potential in helping Smart Manufacturing to achieve effective decision-making based on large amounts of data. Open source software and hardware technologies are getting collaboratively designed and developed for solving the industrial and engineering challenges, including those related to the integration of traditional information technology (IT) systems with OT systems. For example, Bosch IoT Suite is an example of a commercial DT technology that is based on open source software.

Acknowledgements

This research has been partially funded by the Austrian Research Promotion Agency (FFG) and the Austrian Federal Ministry for Transport, Innovation and Technology (BMVIT) within the "ICT of the Future" project IoT4CPS (Trustworthy IoT for Cyber-Physical Systems) (funded for the period between 1 December 2017—20 November 2020). It has been also funded by the European Commission within the H2020 project NIMBLE (Collaborative Network for Industry, Manufacturing, Business and Logistics in Europe), No. 723810, for the period between 1 October 2016 - 31 September 2019.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This research has been partially funded by the Austrian Research Promotion Agency (FFG) and the Austrian Federal Ministry for Transport, Innovation and Technology (BMVIT) within the "ICT of the Future" project IoT4CPS, and by the European Commission within the H2020 project NIMBLE (Collaborative Network for Industry, Manufacturing, Business and Logistics in Europe), No. [723810].

References

Afify, Y., I. F. Moawad, N. Badr, and M. Tolba 2013. "A Semantic-Based Software-As-A-Service (SAAS) Discovery and Selection System." In Proceedings of the 8th IEEE International Conference on Computer Engineering & Systems (IC-CES), Cairo, Egypt, pp. 57–63. doi:10.1177/1753193412446112.

Aggarwal, S., S. Phadke, and M. Bhandarkar 2010. "Characterization of Hadoop Jobs Using Unsupervised Learning." In Proc. of the Cloud Computing Technology and Science (CloudCom), 748–753. doi:10.1109/CloudCom.2010.20.

Alsafi, Y., and V. Vyatkin. 2010. "Ontology-Based Reconfiguration Agent for Intelligent Mechatronic Systems in Flexible Manufacturing." *In Robotics and Computer-Integrated Manufacturing* 26 (4): 381–391. doi:10.1016/j.rcim.2009.12.001.

Ameri, F., C. Urbanovsky, and C. McArthur 2012. "A Systematic Approach to Developing Ontologies for Manufacturing Service Modeling." In Proc. of the 7th Inter. Conf. on Formal Ontology in Information Systems (FOIS). doi:10.1094/PDIS-11-11-0999-PDN.

Ashbrook, D., and T. Starner 2002. "Learning Significant Locations and Predicting User Movement with GPS." In Proc. of the IEEE 6th Inter. Symposium on Wearable Computers (ISWC 2002), doi: 10.1109/ISWC.2002.1167224.

Ashbrook, D., and T. Starner. 2003. "Using GPS to Learn Significant Locations and Predict Movement across Multiple Users." *Personal and Ubiquitous Computing* 7: 275–286. doi:10.1007/s00779-003-0240-0.

Avancha, S., C. Patel, and A. Joshi 2004. "Ontology-Driven Adaptive Sensor Networks. Mobile and Ubiquitous Systems: Networking and Services." In Proceedings of the 1st Annual International Conference on MOBIQUITOUS 2004, Boston, MA, USA, pp. 194–202.

Baheti, R., and H. Gill. 2011. "Cyber-Physical Systems." *The Impact of Control Technology* 12: 161–166.

Bahga, A., and V. K. Madisetti. 2011. "Synthetic Workload Generation for Cloud Computing Applications." *Journal of Software Engineering and Applications* 4: 396–410. doi:10.4236/jsea.2011.47046.

Berman, D. 2017. "InfluxDB Vs. Elasticsearch for Time Series Analysis." https://logz.io/blog/influxdb-vs-elasticsearch/

Boschert, S., and R. Rosen. 2016. "Digital Twin—The Simulation Aspect." In *Mechatronic Futures: Challenges and Solutions for Mechatronic Systems and Their Designers*, edited by P. Hehenberger and D. Bradley, 9–74, Springer International Pub., Cham.

Cai, M., W. Y. Zhang, and K. Zhang. 2001. "ManuHub: A Semantic Web System for Ontology-Based Service Management, Distributed Manufacturing Environments." *In IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans* 41 (3): 574–582. doi:10.1109/TSMCA.2010.2076395.

Chang, X., R. Rai, and J. Terpenny. 2010. "Development and Utilization of Ontologies in Design for Manufacturing." *Journal on Mechanical Design* 132 (2): 021009–021009-12,1–12. doi:10.1115/1.4000697.

Chen, Y., A. Ganapathi, R. Griffith, and R. H. Katz 2010. "Analysis and Lessons from a Publicly Available Google Cluster Trace." In Proceedings of the EECS Department, University of California, Berkeley.

Choi, S. S., T. H. Yoon, and S. D. Noh. 2010. "XML-based Neutral File and PLM Integrator for PPR Information Exchange between Heterogeneous PLM Systems." *International Journal of Computer Integrated Manufacturing* 23 (3): 216–228. doi:10.1080/09511920903443234.

Chungoora, N., R. I. Young, G. Gunendran, C. Palmer, Z. Usman, N. A. Anjum, A.-F. Cutting-Decelle, J. A. Harding, and K. Case. 2013. "A Model-Driven Ontology Approach for Manufacturing System Interoperability and Knowledge Sharing." *Journal on Computers in Industry* 64 (4): 392–401. doi:10.1016/j.compind.2013.01.003.

Churilo, C. 2018. "InfluxDB Vs. Elasticsearch for Time Series Data & Metrics Benchmark." https://bit.ly/2snLUN5

Cimini, C., R. Pinto, and S. Cavalieri 2017. "The Business Transformation Towards Smart Manufacturing: A Literature Overview about Reference Models and Research Agenda." In Proceedings of the International Federation of Automatic Control Conference (IFAC). 1452–1457. doi: 10.1016/j.ifacol.2017.08.2548.

Damjanovic-Behrendt, V. 2018. "A Digital Twin Architecture for Security, Privacy and Safety". ERCIM News No. 115, Special Issue "Digital Twins". https://bit.ly/2CCU0Il

Davis, J., T. Edgar, J. Porter, J. Bernaden, and M. Sarli. 2012. "Smart Manufacturing, Manufacturing Intelligence and Demand-Dynamic Performance." *Computers & Chemical Engineering* 47: 145–156. doi:10.1016/j.compchemeng.2012.06.037.

Diep, D., C. Alexakos, and T. Wagner 2007. "An Ontology-Based Interoperability Framework for Distributed Manufacturing Control." In Proceedings of the IEEE Conference on Emerging Technologies and Factory Automation (ETFA), Patras, Greece, 855–862. doi:10.1094/PDIS-91-4-0467B.

Dobbelaere, P., and K. S. Esmaili 2017. "Kafka versus RabbitMQ: A Comparative Study of Two Industry Reference Publish/Subscribe Implementations: Industry Paper." In Proceedings of the 11th ACM International Int. Conf. on Distributed and Event-based Systems (DEBS '17), 227–238. doi:10.2147/COPD.S130231.

DZone. 2018. "DZone Guide to Open Source: Democratizing Development." Accessed May 2018. https://bit.ly/2jLCN4V

ECLIF. 2010. "Extended Common Logic Interchange Format (ECLIF) Reference." In: Proprietary document supplied with the installation of the Highfleet Integrated Ontology Development Environment (IODE) Application.

Esposito, C., A. Castiglione, and K.-K. R. Choo. 2016. "Challenges in Delivering Software in the Cloud as Microservices." *IEEE Cloud Computing* 3 (5): 10–14. doi:10.1109/MCC.2016.105.

Fan, J., F. Han, and H. Liu. 2014. “Challenges of Big Data Analysis.” *National Science Review* 1 (2): 293–314. doi:10.1093/nsr/nwt032.

Freeman, H. 2016. “Streaming Analytics 101: The What, Why, and How.” http://www.dataversity.net/streaming-analytics-101/

Gabor, T., L. Belzner, and M. Kiermeier 2016. “A Simulation-Based Architecture for Smart Cyber-Physical Systems.” In Proceedings of the 2016 IEEE International Conference on Autonomic Computing (ICAC), 2016:374–379. doi: 10.1109/ICAC.2016.29.

Gandomi, A., and M. Haider. 2015. “Beyond the Hype: Big Data Concepts, Methods, and Analytics.” *International Journal of Information Management* 35 (2): 137–144. doi:10.1016/j.ijinfomgt.2014.10.007.

Garetti, M., and L. Fumagalli. 2012. “P-PSO Ontology for Manufacturing Systems.” *INCOM 2012—Information Control in Manufacturing Conference* 14 (1): 449–456. doi:10.3182/20120523-3-RO-2023.00222.

Garetti, M., and L. Fumagalli 2012a. “Role of Ontologies in Open Automation of Manufacturing Systems.” In Proceedings of the 17th Summer School in Industrial Mechanical Plants, Venice, Italy.

Garetti, M., L. Fumagalli, A. Lobov, and J. Martinez Lastra. 2013. “Open Automation of Manufacturing Systems through Integration of Ontology and Web Services.” In Proceedings of the 7th IFAC Conference on Manufacturing Modelling, Management, and Control. doi:10.3182/20130619-3-RU-3018.00169.

Garetti, M., L. Fumagalli, and E. Negri. 2015. “Role of Ontologies for CPS Implementation in Manufacturing.” *In Management and Production Engineering Review* 6: 4. doi:10.1515/mper-2015-0033.

Geng, X., H. Arimura, and T. Uno 2012. “Pattern Mining from Trajectory GPS Data.” In Proceedings of the 2012 IIAI International Conference on Advanced Applied Informatics, IIAIAAI 2012, Fukuoka, Japan, pp.60–65.

Giovannini, A., A. Aubry, H. Panetto, M. Dassisti, and H. E. Haouzi. 2012. “Ontology-Based System for Supporting Manufacturing Sustainability.” *Annual Reviews in Control* 36 (2): 309–317. doi:10.1016/j.arcontrol.2012.09.012.

Grieves, M. 2014. “Digital Twin: Manufacturing Excellence through Virtual Factory Replication.” White paper. https://bit.ly/2slbtiR

Grieves, M., and J. Vickers. 2017. “Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems.” In *Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches*, edited by F.-J. Kahlen, S. Flumerfelt, and A. Alves, 85–113. Springer International Pub. doi:10.1007/978-3-319-38756-7_4.

Haag, S., and R. Anderl. 2018. “Digital Twin - Proof of Concept.” *Manufacturing Letters* 15 (B): 64–66. doi:10.1016/j.mfglet.2018.02.00.

Horváth, I., and B. Gerritsen. 2012. “Cyber-Physical System: Concepts, Technologies and Manifestation.” *In Proceedings of the TMCE 2012* 1: 1–16.

Industrial Internet Consortium. 2017. “The Industrial Internet of Things.” Volume G1: Reference Architecture. https://bit.ly/2B1Lteg

IoT Eclipse.org, 2017. “Open Source Software for Industry 4.0.” Accessed February 2019. https://bit.ly/2gIbVow

Jazdi, N. 2014. “Cyber Physical Systems in the Context of Industry 4.0.” In Proceeding of the 2014 IEEE International Conference on Automation, Quality and Testing, Robotics. pp.1–4. doi: 10.1109/AQTR.2014.6857843.

Kang, H. S., J. Y. Lee, S. Choi, H. Kim, J. H. Park, J. Y. Son, B. H. Kim, and S. D. Noh. 2016. “Smart Manufacturing: Past Research, Present Findings, and Future Directions.” *International Journal of Precision Engineering and Manufacturing-Green Technologies* 3: 111–128. doi:10.1007/s40684-016-0015-5.

Kiritsis, D. 2011. “Closed-Loop PLM for Intelligent Products in the Era of the Internet of Things.” *Computer-Aided Design* 43 (5): 479–501. doi:10.1016/j.cad.2010.03.002.

Kiritsis, D., S. Kadiri, A. Perdikakis, A. Milicic, D. Alexandrou, and K. Pardalis. 2013. “Design of Fundamental Ontology for Manufacturing Product Lifecycle Applications.” In *Advances in Production Management Systems. Competitive Manufacturing for Innovative Products and Services. APMS 2012. IFIP Advances in Information and Communication Technology*, edited by C. Emmanouilidis, M. Taisch, and D. Kiritsis. Vol. 397, pp. 376-382. Berlin, Heidelberg: Springer.

Koren, Y., and M. Shpitalni. 2010. “Design of Reconfigurable Manufacturing Systems.” *Journal of Manufacturing Systems* 29 (4): 130–141. doi:10.1016/j.jmsy.2011.01.001.

Koren, Y., U. Heisel, F. Jovane, T. Moriwaki, G. Pritschow, A. G. Ulsoy, and H. Van Brussel. 1999. “Reconfigurable Manufacturing Systems.” *Annals of the CIRP* 48 (2): 527–540. doi:10.1016/S0007-8506(07)63232-6.

Kraft, E. 2016. “The Air Force Digital Thread/Digital Twin - Life Cycle Integration and Use of Computational and Experimental Knowledge.” In Proc. of the 54th AIAA Aerospace Sci. Meeting, AIAA SciTech Forum. doi: 10.2514/6.2016-0897.

Kucera, R., M. Aanenson, and M. Benson, 2017. “The Augmented Digital Twin: Combining Physical and Virtual Data to Unlock the Value of IoT.” Exosite White paper. https://bit.ly/2sirAws

Lee, E. A. 2008. “Cyber Physical Systems: Design Challenges.” In Proceedings of the 11th IEEE Intern. Symposium on Object and Component-Oriented Real-Time Distributed Computing, 363–369. doi: 10.1109/ISORC.2008.25.

Lee, J., B. Bagheri, and H.-A. Kao. 2015. “A Cyber-Physical Systems Architecture for Industry 4.0-Based Manufacturing Systems.” *Manufacturing Letters* 3: 18–23. doi:10.1016/j.mfglet.2014.12.001.

Lee, J., H.-A. Kao, and S. Yang. 2014. “Service Innovation and Smart Analytics for Industry 4.0 And Big Data Environment.” *In Proceedings of CIRP* 16: 3–8. Elsevier. doi:10.1016/j.procir.2014.02.001.

Lemaignan, S., A. Siadat, D. Jean-Yves, and A. Semenenko 2006. “MASON: A Proposal for an Ontology of Manufacturing Domain.” IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications (DIS 2006), Prague, Czech Republic, 195–200.

Lewis, J., and M. Fowler 2014. “Microservices.” Accessed February 2019 https://martinfowler.com/articles/microservices.html

Liao, Y., F. Deschamps, R. E. Loures, and P. F. Ramos. 2017. “Past, Present and Future of Industry 4.0—A Systematic Literature Review and Research Agenda Proposal.” *International Journal of Production Research* 55 (12): 3609–3629. doi:10.1080/00207543.2017.1308576.

Lin, S.-W., et al. 2017. “Architecture Alignment and Interoperability.” An Industrial Internet Consortium and Plattform Industrie 4.0 Joint Whitepaper. IIC:WHT:IN3: V1.0:PB:20171205

Liu, L., X. Yao, L. Qin, and M. Zhang 2014. “Ontology-Based Service Matching in Cloud Computing.” In Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ- IEEE), pp.2544–2550. doi:10.1109/FUZZ-IEEE.2014.6891698.

Lüder, A., L. Hundt, and A. Keibel 2010. “Description of Manufacturing Processes Using AutomationML.” In Proceedings of the 2010 IEEE Conference on Emerging Technologies and Factory Automation, 1–8. doi: 10.1109/ETFA.2010.5641346.

MacDougall, W. 2014. “Industrie 4.0 Smart Manufacturing for the Future.” Mechanical & Electronic Technologies, Germany Trade & Invest (4) 40. Accessed February 2019. https://bit.ly/2RFdbW9

McCool, M. 2017. “The Web of Things and Semantic Interoperability.” IETF1000. Accessed February 2019. https://bit.ly/2GisJ0M

McLeod, I., H. Yu, and E. Mahdi. 2011. “Time Series Analysis with R.” *Elsevier Handbook of Statistics* 30. Available https://bit.ly/2L9J2uY

Menzel, C., and M. Grüninger 2001. “A Formal Foundation for Process Modeling.” In Proceedings of the Second International Conference on Formal Ontologies in Information Systems, Welty and Smith (Eds.), 256–269.

Mishra, A., J. Hellerstein, W. Cirne, and C. Das. 2010. “Towards Characterizing Cloud Backend Workloads: Insights from Google Compute Clusters.” *In SIGMETRICS Performance Evaluation Review* 37, Fukuoka, Japan. pp. 34–41. doi:10.1145/1773394.1773400.

Mnih, V., H. Larochelle, and G. E. Hinton 2011. “Conditional Restricted Boltzmann Machines for Structured Output Prediction.” In Proceedings of the 27th Conference on Uncertainty in AI (UAI’11), F. Cozman and A. Pfeffer (Eds.). AUAI Press, 514–522. doi:10.1177/1753193411406479.

Mohammed, W. M., B. Ramis Ferrer, S. Iarovyi, E. Negri, L. Fumagalli, A. Lobov, and J. Martinez Lastra. 2017. “Generic Platform for Manufacturing Execution System Functions in Knowledge-Driven Manufacturing Systems.” *International Journal of Computer Integrated Manufacturing* 31 (3): 262–274.

Möller, D. P. F. 2016. “Digital Manufacturing/Industry 4.0..” In *Guide to Computing Fundamentals in Cyber-Physical Systems*, 307–375.

Monostori, L. 2014. “Cyber-Physical Production Systems: Roots, Expectations and R&D Challenges.” *In Proceedings of the 47th CIRP Conference on Manufacturing Systems* 17: 9–13.

Negri, E., L. Fumagalli, M. Macchi, and M. Garetti 2015a. “Ontology for Service-Based Control of Production Systems.” In S. Umeda, M. Nakano, H.

Mizuyama, H. Hibino, D. Kiritsis, and G. von Cieminski (Eds.) IFIP Int. Conf. on Advances in Production Management Systems (APMS), Japan, 484–492.

Negri, E., L. Fumagalli, M. Garetti, and L. Tanca. 2015. "Requirements and Languages for the Semantic Representation of Manufacturing Systems." *Computers in Industry* 81 (C): 55–66. doi:10.1016/j.compind.2015.10.009.

Negri, E., L. Fumagalli, and M. Macchi 2017. "A Review of the Roles of Digital Twin in CPS-based Production Systems." In Proceedings of the 27th Int. Conf. on Flexible Automation and Intelligent Manufacturing (FAIM2017), Italy. Vol 11, 939–948.

Negri, E., S. Perotti, L. Fumagalli, G. Marchet, and M. Garetti. 2017. "Modelling Internal Logistics Systems Through Ontologies." *Journal on Computers in Industry* 88 (C): 19–34. doi:10.1016/j.compind.2017.03.004.

Neuhaus, H., and M. Compton 2009. "The Semantic Sensor Network Ontology: A Generic Language to Describe Sensor Assets." In Proceedings of the AGILE Workshop: Challenges in Geospatial Data Harmonization, hannover, Germany.

NIST Statistics. 2012. "NIST Engineering Statistics Handbook. Introduction to Time Series Analysis." Accessed May 2018. https://bit.ly/2H0CQTv

oneM2M. 2018, March. "oneM2M Technical Specification: Base Ontology."

OPC Foundation. 2016. "OPC Unified Architecture: Interoperability for Industrie 4.0 And the Internet of Things." https://bit.ly/29hTRZw

Oracle. 2017. "Oracle: Digital Twin for IoT Applications: A Comprehensive Approach to Implementing IoT Digital Twin." Oracle White Paper. https://bit.ly/2IZfFyY

Oussous, A., F.-Z. Benjelloun, A. Ait Lahcen, and S. Belfkih. 2017. "Big Data Technologies: A Survey." *Journal of King Saud University - Computer and Information Sciences*, 30 (4): 431-448. http://dx.doi.org/10.1016/j.jksuci.2017.06.001

Overton, J., and J.-C. Brigham. 2017. "The Digital Twin. Data-Driven Simulations Innovate the Manufacturing Process." *White paper*. Accessed 26 May 2018 https://bit.ly/2xton2W

Prasad, B. R., and S. Agarwal. 2016. "Comparative Study of Big Data Computing and Storage Tools: A Review." *International Journal of Database Theory and Application* 9: 45–66. doi:10.14257/ijdta.2016.9.1.05.

Pratt, M. 2001. "Introduction to ISO10303 the STEP Standard for Product Data Exchange." *Journal of Computing and Information Science in Engineer* 1 (1): 102–103. doi:10.1115/1.1354995.

Predix. 2018. "PREDIX the Industrial IoT Application Platform." Technical Brief. Accessed 26 May 2018. https://invent.ge/2LCgwmO

PTC Windchill. 2018. "Windchill, from Requirements to Quality and from Manufacturing to Services." https://bit.ly/2LM4BDd

Qiu, W., and A. Bandara 2015. "GPS Trace Mining for Discovering Behavior Patterns." In Proceedings of the 2015 International Conference on Intelligent Environments. IEEE Computer Society, USA, pp. 65–72. doi:10.1109/IE.2015.17.

Rajkumar, R., I. Lee, L. Sha, and J. Stankovic 2010. "Cyber-Physical Systems: The Next Computing Revolution." In Proceedings of the 47th Design Automation Conference (DAC), Anaheim, California, USA, pp. 731–736.

Rios, J., J. C. Hernandez, M. Oliva, and F. Mas 2015. "Product Avatar as Digital Counterpart of a Physical Individual Product: Literature Review and Implications in an Aircraft System." In Proceedings of the 22nd International Conference on Concurrent Engineering (ISPE CE2015), Vol. 2 of Advances in Transdisciplinary Engineering, 657–666. doi: 10.3233/978-1-61499-544-9-657.

Rodríguez-García, M. Á., R. Valencia-García, F. García-Sánchez, and J. J. Samper-Zapater. 2014. "Ontology-Based Annotation and Retrieval of Services in the Cloud." *Knowledge-Based Systems* 56 (C): 15–25. doi:10.1016/j.knosys.2013.10.006.

Rosen, R., G. von Wichert, G. Lo, and K. D. Bettenhausen. 2015. "About the Importance of Autonomy and Digital Twins for the Future of Manufacturing." *In Proceedings of the 15th IFAC Symposium on Information Control Problems in Manufacturing (INCOM 2015)* 48 (3): 567–572.

Russomanno, D., C. Kothari, and O. Thomas. 2005. "Building A Sensor Ontology: A Practical Approach Leveraging ISO and OGC Models." In Proceedings of the 2005 International Conference on Artificial Intelligence (ICAI 2005), Las Vegas, Nevada, USA. Vol. 2, pp. 637–643.

Sarma, S., D. L. Brock, and K. Ashton 2000. "The Networked Physical World." Proposals for Engineering the Next Generation of Computing, Commerce & Automatic-Identification. White paper. https://bit.ly/2sknSU5

Schleich, B., N. Anwer, L. Mathieu, and S. Wartzack. 2017. "Shaping the Digital Twin for Design and Production Engineering." *In CIRP Annals, Manufacturing Technol.* 6 (1): 141–144. doi:10.1016/j.cirp.2017.04.040.

Schlenoff, C., M. Grüninger, M. Ciocoiu, and J. Lee. 2000. "The Essence of the Process Specification Language." *Transactions of the Society for Computer Simulation* 16 (4) :204–216.

Schluse, M., and J. Rossmann 2016. "From Simulation to Experimentable Digital Twins - Simulation-Based Development and Operation of Complex Technical Systems." In Proceedings of the 2nd IEEE International Symposium on Systems Engineering (ISSE 2016), pp. 1–6, doi: 10.1109/SysEng.2016.7753162.

Schroeder, G., C. Steinmetz, C. Pereira, and D. Espindola. 2016. "Digital Twin Data Modeling with AutomationML and a Communication Methodology for Data Exchange." *In International Federation of Automatic Control Symposium on Telematics Applications (TA 2016)* 49 (30): 12–17. doi:10.1016/j.ifacol.2016.11.115.

Serpanos, D. 2018. "The Cyber-Physical Systems Revolution." *In Computer* 51 (3): 70–73. doi:10.1109/MC.2018.1731058.

Shafto, M., M. Conroy, R. Doyle, E. Glaessgen, C. Kemp, J. LeMoigne, and L. Wang. 2010. "DRAFT Modeling, Simulation, Information Technology & Processing Roadmap. Technology Area 11".NASA. Online: https://www.nasa.gov/pdf/501321main_TA11-MSITP-DRAFT-Nov2010-A1.pdf

Siemens Simcenter. 2017. "Predict Performance of Your 3D Geometry-Based Design." https://sie.ag/2xxfWUA

Smith, J. W., and I. Sommerville 2011. "Workload Classification & Software Energy Measurement for Efficient Scheduling on Private Cloud Platforms." In Proceedings of the ACM SOCC 2011. Available: https://arxiv.org/pdf/1105.2584.pdf

Sutton, R., and A. Barto. 2012. *Reinforcement Learning: An Introduction*. Cambridge: MIT Press.

Szilagyi, I., and P. Wira 2016. "Ontologies and Semantic Web for the Internet of Things - A Survey." In Proceedings of the 42nd IEEE Industrial Electronics Society Conference (IECON 2016), Florence, Italy, pp. 6949-6954. doi:10.1109/IECON.2016.7793744).

Taibi, D., V. Lenarduzzi, and C. Pahl. 2017. "Processes, Motivations, and Issues for Migrating to Microservices Architectures: An Empirical Investigation." *IEEE Cloud Computing* 4 (5): 22–32. doi:10.1109/mcc.2017.4250931.

Tao, F., H. Zhang, A. Liu, and A. Y. C. Nee. 2019. "Digital Twin in Industry: State-of-the-art." *IEEE Transactions on Industrial Informatics* 15 (4): 2405-2415. doi:10.1109/TII.2018.2873186.

Tao, F., J. Cheng, Q. Qi, M. Zhang, H. Zhang, and F. Sui. 2017. "Digital Twin-Driven Product Design, Manufacturing and Service with Big Data." *International Journal of Advanced Manufacturing Technology* 94 (9–12): 3563–3576. doi:10.1007/s00170-017-0233-1.

Tavčar, J., and I. Horváth. 2018. "A Review of the Principles of Designing Smart Cyber-Physical Systems for Run-Time Adaptation: Learned Lessons and Open Issues." *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. doi:10.1109/TSMC.2018.2814539.

Taylor, G. W., and G. E. Hinton 2009. "Factored Conditional Restricted Boltzmann Machines for Modelling Motion Style." In Proc. of the 26th Annual International Conference on Machine Learning (ICML '09). ACM, 1025–1032. doi:10.1016/j.actbio.2009.09.014.

Tsoukatos, I., and D. Gunopulos 2001. "Efficient Mining of Spatiotemporal Patterns." In Proceedings of the International Symposium on Spatial and Temporal Databases (SSTD), Redondo Beach, Los Angeles, CA.pp. 425-442.

Uhlemann, T. H.-J., C. Schock, C. Lehmann, S. Freiberger, and R. Steinhilper. 2017. "The Digital Twin: Demonstrating the Potential of Real Time Data Acquisition in Production Systems." *In Proceedings of the 7th Conference on Learning Factories, CLF 2017* 9: 113–120. doi:10.1016/j.promfg.2017.04.043.

Usha, D., and A. P. S. Aslin Jenil. 2014. "A Survey of Big Data Processing in Perspective of Hadoop and MapReduce." *International Journal of Current Engineering and Technology*, Vol 4 (2), pp.602-606.

Von Euler-Chelpin, A. 2008. "Information Modelling for the Manufacturing System Life Cycle." Doctoral Thesis, KTH Royal Institute of Technology, Sweden.

W3C TAG. 2018. "Guidelines for Creating Web Platform Compatible Components." W3C Technical Architecture Group (TAG). Accessed

February 2019. https://www.w3.org/2001/tag/doc/webcomponents-design-guidelines/

Wang, S., J. Wan, D. Li, and C. Zhang. 2016. "Implementing Smart Factory of Industrie 4.0: An Outlook." *International Journal of Distributed Sensor Networks*, no. 4: 1–10. doi:10.1155/2016/3159805.

Weyer, S., T. Meyer, M. Ohmer, D. Gorecky, and D. Zühlke 2016. "Future Modeling and Simulation of CPS-based Factories: An Example from the Automotive Industry." In Proceedings of the IFAC (International Federation of Automatic Control) Conference, 97–102. doi:10.1177/1753193415618128.

White, T. 2012. *Hadoop: The Definitive Guide*. O'Reilly Media, Inc., Sebastopol, CA, USA.

Yang, H., Z. Luan, W. Li, and D. Qian. 2012. "MapReduce Workload Modeling with Statistical Approach." *Journal on Grid Computing* 10 (2): 279–310. doi:10.1007/s10723-011-9201-4.

Yoon, J.-S., S.-J. Shin, and S.-H. Suh. 2012. "A Conceptual Framework for the Ubiquitous Factory." *International Journal of Production Research* 50 (8): 2174–2189. doi:10.1080/00207543.2011.562563.

Young, R. I. M., A. G. Gunendran, A. F. Cutting-Decelle, and M. Gruninger. 2007. "Manufacturing Knowledge Sharing in PLM: A Progression Towards the Use of Heavy Weight Ontologies." *International Journal of Production Research* 45 (7): 1505–1519. doi:10.1080/00207540600942268.

Zhang, P., J. K. Liu, F. R. Yu, M. Sookhak, M. H. Au, and X. Luo. 2018. "A Survey on Access Control in Fog Computing." *IEEE Communications Magazine* 56 (2): 144–149. doi:10.1109/MCOM.2018.1700333.

Zuehlke, D. 2010. "SmartFactory - Towards a Factory-of-Things." *In Annual Review in Control* 34: 129–138. doi:10.1016/j.arcontrol.2010.02.008.

ZVEI. 2015. "Reference Architectural Model Industrie 4.0 (RAMI 4.0)." Status Report. https://bit.ly/2L66o4J