DOCTORAL THESIS

UNIVERSITÀ DEGLI STUDI DELL'INSUBRIA

PH.D. DEGREE PROGRAM IN
COMPUTER SCIENCE AND COMPUTATIONAL MATHEMATICS

AN INTEGRATIVE FRAMEWORK FOR COOPERATIVE PRODUCTION RESOURCES IN SMART MANUFACTURING

DR. GIANFRANCO EMANUELE MODONI

Advisor	Co-Advisor
Prof. Alberto TROMBETTA	Dr. Marco SACCO
Università degli studi dell'Insubria	CNR-ITIA

Academic Year 2016-2017

“Machines take me by surprise with great frequency.”
A. M. Touring

To my wife Marcella
To my son Francesco Maria
To my father Totò and my mother Maria Teresa
To my brother Mario

Contents

List of Figures

Abstract

Industry 4.0 paradigm envisions a new generation of collaborative manufacturing system where all the components are connected exploiting the Industrial Internet of Things (IIoT) protocol. Through this pervasive connection, sensors, machines, robots and other productive equipments can communicate and share information with their surroundings resources. In order to contribute to realize such a vision, this doctoral thesis work explores a cooperative pattern for the interaction of distributed heterogeneous resources in the field of manufacturing, according to the principles of Industry 4.0 paradigm. The overall aim is providing new mechanisms to propagate new information produced within the factory towards all the enabled and interested production resources. The idea behind this research work goes in the direction to explore and identify scalable technological solutions to answer the continued growth of smart objects connected to the IIoT network. Through the combination of various technologies ranging from IIoT to Big Data, from Semantic Web to Multi-Agent based systems, the proposed model of collaboration will allow the resources to be updated about changes occurred in their context and then apply some actions in response to business or operational variations. Under these conditions, the new model of collaboration can contribute to support various ad-hoc services that accompany modern factories towards a more sustainable, efficient, and competitive manufacturing system. The opportunity to follow these studies driven by real case studies under the Italian Design For All project and under the ongoing European H2020 INTER-IoT project makes this work more interesting and challenging. With regards to INTER-IoT project, the submitted proposal, whose I am main author, is mainly based on the application of preliminary outcomes achieved during my Phd. The quality and the correctness of the proposed approach is confirmed by the fact that proposal resulted one of the winners of the first INTER-IoT Open Call.

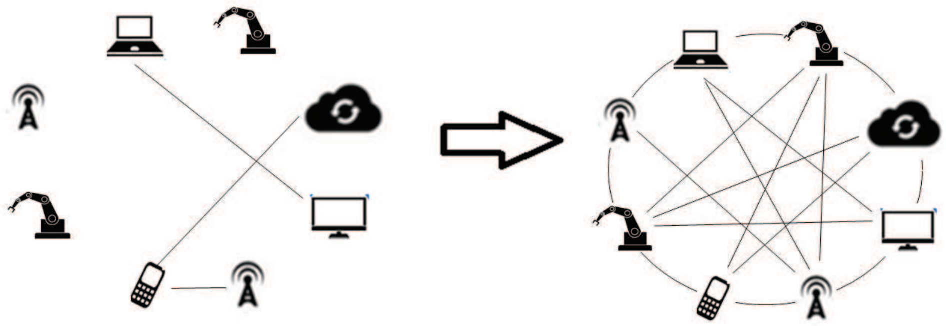
Chapter 1

Research overview and problem statement

1.1 Introduction

Under the push of Industry 4.0 paradigm modern manufacturing companies are dealing with a significant digital transition, with the aim to better address the challenges posed by the growing complexity of globalized businesses (Hermann, Pentek, and Otto 2016). One basic principle of this paradigm is that products, machines, systems and business are always connected to create an intelligent network along the entire factory's value chain. According to this vision, manufacturing resources are being transformed from monolithic entities into distributed components, which are loosely coupled and autonomous but nevertheless provided of the networking and connectivity capabilities enabled by the increasingly widespread Industrial Internet of Things technology (fig. 1.1). Under these conditions, they become capable of working together in a reliable and predictable manner, collaborating among themselves in a highly efficient way. Such a mechanism of synergistic collaboration is crucial for the correct evolution of any organization ranging from a multi-cellular organism to a complex modern manufacturing system (Moghaddam and Nof 2017). Specifically of the last scenario, which is the field of our study, collaboration enables involved resources to exchange relevant information about the evolution of their context. These information can be in turn elaborated to make some decisions, and trigger some actions. In this way connected resources can modify their structure and configuration in response to specific business or operational variations (Alexopoulos et al. 2016). Such a model of "social" and context-aware resources can contribute to the realization of a highly flexible, robust and responsive manufacturing system, which is an objective particularly relevant in the modern factories, as its inclusion in the scope of the priority research lines for the H2020 three-year period 2018-2020 can demonstrate (EFFRA 2016). Interesting examples of these resources are self-organized logistics which can react to unexpected changes occurred in production or machines capable to predict failures

Figure 1.1: Evolution of the manufacturing resources into IIoT connected resources



on the basis of the contextual information and then trigger adjustments processes autonomously.

This vision of collaborative and cooperative resources can be realized with the support of several studies in various fields ranging from information and communication technologies to artificial intelligence. An updated state of the art highlights significant recent achievements that have been making these resources more intelligent and closer to the user needs. However, we are still far from an overall implementation of the vision, which is hindered by three major issues. The first one is the limited capability of a large part of the resources distributed within the shop floor to automatically interpret the exchanged information in a meaningful manner (semantic interoperability) (Atzori, Iera, and Morabito 2010). This issue is mainly due to the high heterogeneity of data model formats adopted by the different resources used within the shop floor (Modoni et al. 2017a). Another open issue is the lack of efficient methods to fully virtualize the physical resources (Rosen et al. 2015), since only pairing physical resource with its digital counterpart that abstracts the complexity of the real world, it is possible to augment communication and collaboration capabilities of the physical component. The third issue is a side effect of the ongoing technological ICT evolutions affecting all the manufacturing companies and consists in the continuous growth of the number of threats and vulnerabilities, which can both jeopardize the cyber-security of the overall manufacturing system (Wells et al. 2014). For this reason, aspects related with cyber-security should be considered at the early stage of the design of any ICT solution, in order to prevent potential threats and vulnerabilities. All three of the above mentioned open issues have been addressed in this research work with the aim to explore and identify a precise, secure and efficient model of collaboration among the production resources distributed within the shop floor.

This document illustrates main outcomes of the research, focusing mainly on the Virtual Integrative Manufacturing Framework for resources Interaction (VICKI), a potential reference architecture for a middleware application enabling semantic-based cooperation among manufacturing resources. Specifically, this framework provides a technological and service-oriented infrastructure offering an event-driven mechanism that dynamically propagates the changing factors to the interested devices. The proposed system supports the coexistence and combination of physical components and their virtual counterparts in a network of interacting collaborative elements in constant connection, thus allowing to bring back the manufacturing system to a cooperative Cyber-physical Production System (CPPS) (Monostori 2014). Within this network, the information coming from the productive chain can be promptly and seamlessly shared, distributed and understood by any actor operating in such a context. In order to overcome the problem of the limited interoperability among the connected resources, the framework leverages a common data model based on the Semantic Web technologies (SWT) (Berners-Lee, Hendler, and Lassila 2001). The model provides a shared understanding on the vocabulary adopted by the distributed resources during their knowledge exchange. In this way, this model allows to integrate heterogeneous data streams into a coherent semantically enriched scheme that represents the evolution of the factory objects, their context and their smart reactions to all kind of situations. The semantic model is also machine-interpretable and re-usable. In addition to modeling, the virtualization of the overall manufacturing system is empowered by the adoption of an agent-based modeling, which contributes to hide and abstract the control functions complexity of the cooperating entities, thus providing the foundations to achieve a flexible and reconfigurable system. Finally, in order to mitigate the risk of internal and external attacks against the proposed infrastructure, it is explored the potential of a strategy based on the analysis and assessment of the manufacturing systems cyber-security aspects integrated into the context of the organization's business model.

To test and validate the proposed framework, a demonstration scenarios has been identified, which are thought to represent different significant case studies of the factory's life cycle. To prove the correctness of the approach, the validation of an instance of the framework is carried out within a real case study. Moreover, as for data intensive systems such as the manufacturing system, the quality of service (QoS) requirements in terms of latency, efficiency, and scalability are stringent, an evaluation of these requirements is needed in a real case study by means of a defined benchmark, thus showing the impact of the data storage, of the connected resources and of their requests.

1.2 Vision

Manufacturing companies are projected into a new era in which they will be capable to exploit the IIoT in its full potential. In this regard, the ongoing advance of this technology, also in combination with other enabling technologies (e.g. Big Data, etc.), will play a crucial role for the development of the factories of the future, with the overall realization of the paradigms of the Smart, Digital and Virtual Manufacturing. Leveraging the IIoT, manufacturing systems will become dynamic networks comprising different kinds of physical and virtual resources. The physical resources, representing new or legacy upgraded machinery, will become more intelligent, thanks to their advanced communication and networking capabilities which will allow these resources to actively cooperate and collaborate to provide smart functionalities.

In this way, objects, machines and robots will form a sort of "nervous system" within the factory world (Vermesan and Friess 2015). On one hand these capabilities will enable internal and external access to information handled by the working resources (machines and other devices), while in traditional manufacturing systems the access to this information is typically hidden in closed silos. The other hand, these capabilities will connect the real factory to a broad set of internal and external services and applications (e.g. the smart grid services, the facilities services within the production, etc.) (fig. 1.2). In the IIoT network, production resources ("things") will interact with each other by exchanging information regarding their status, properties and history, reacting independently to events by performing processes that trigger actions and create services, even without direct human intervention. Under these conditions, the critical components of a manufacturing system can become more context-aware, interactive and efficient, becoming active participants in business processes.

The existence of resources connected anytime, anyplace and with anything will also allow to overcome the hierarchical organization of the shop-floor based on the pyramid automation, being each one of these intelligent resources capable of providing complex services across all the layers. Under these conditions, there will be an evolution and a reshape of the pyramid representation to take into account the service-oriented architecture of the intelligent resources (fig. 1.3). By overcoming the hierarchical factory automation pyramid, services, applications and any other component of the factories of the future does not have to be developed in a strictly linked way to the physical system, but it can be rather specified as services in a shared physical world.

Figure 1.2: The Internet of Things ecosystems

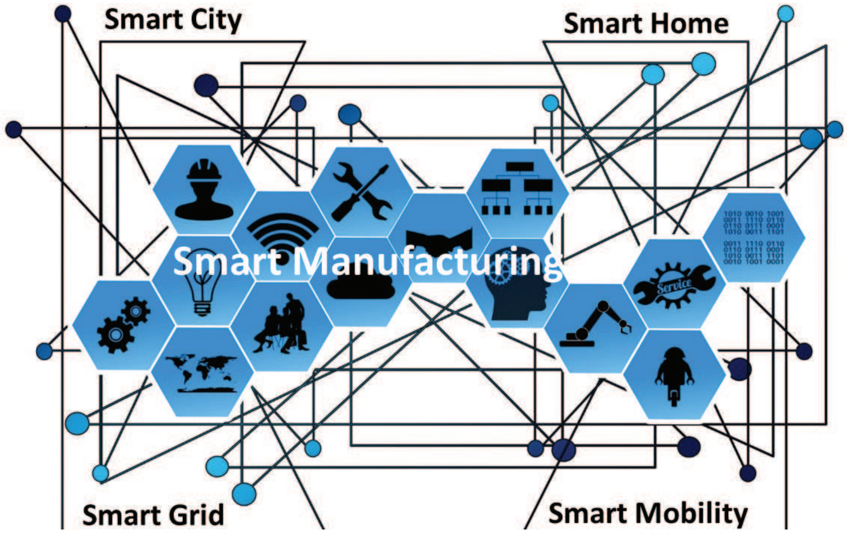
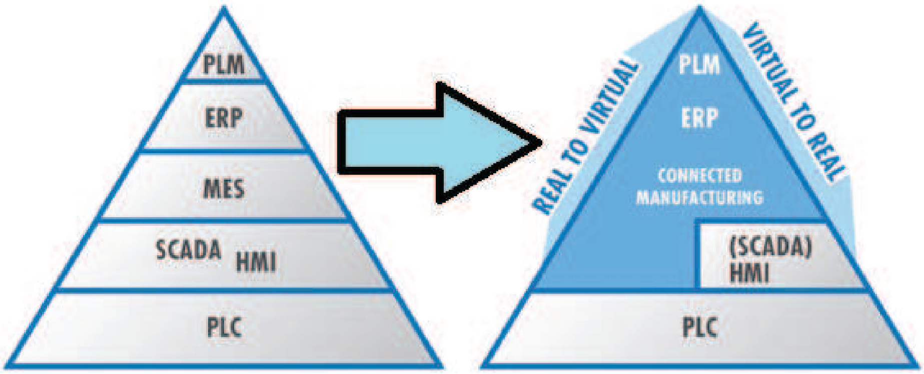


Figure 1.3: Evolution of the classic Automation Pyramid



1.3 Motivations

Today's manufacturing systems consist in a broad collection of various hardware and software components, which range from different CNC machines, robots and PLCs to complex software applications. These components, distributed and connected through various communication protocols, are typically responsible for one or more specific activities. Each of them produces worthwhile information about different aspects of their mechanical machining. These information can be in turn exploited by other resources in order to elicit new strategic knowledge concerning the whole factory world. Indeed, as manufacturing processes are flows of activities linked each other, the information produced during any production process can be relevant inputs of a subsequent process along the whole production and/or along the supply chain.

In traditional manufacturing systems information are typically shared through a communication protocols that are proprietary, not standardized and based on heterogeneous data formats. In such a context, the communication and collaboration capabilities of these resources are limited in vertical closed systems. As a result, current resources have only a partial fragmented view of the data produced by other resources. For the same reasons, they do not have an efficient access to an holistic view aggregating data produced from the overall system. Thus, the huge amount of relevant information produced within the shop floor are underused for gaining relevant insights in near real-time, e.g. detecting production problems such as faults and failures. Under these conditions, it is crucial to explore new strategies to interconnect and synchronize the productive resources, in particular enabling effective and efficient mechanisms for the exchange of relevant data. Since for any complex organization, collaboration and information sharing is an essential mechanism for its evolution (Moghaddam and Nof 2017), a suitable environment is needed in which different components can run and interact, establishing with each-other agreements for their initialization, synchronization, events dispatching and termination. The motivation of this work lies in understanding how to conceive, design and realize such an environment and how to integrate it with all kind of physical resources (also legacy machines), taking also into account issues such as the virtualization of the real resources and the enforcement of their cyber-security. The opportunity to implement this system for real case studies under the Italian Design For All project (closed on February 2017) (Sacco et al. 2014) and under the ongoing European H2020 INTER-IoT project (First Open Call) (*INTER-IoT. Online*) makes this work more interesting and challenging. With regards to INTER-IoT project, the submitted proposal, whose I am main author, is mainly based on the application of preliminary outcomes achieved during my Phd. The quality and the correctness of the proposed approach is confirmed by the fact that proposal resulted one of the winners of the first INTER-IoT Open Call.

To support manufacturing companies in the challenge to overcome the current fragmentation, the main objective of this research-work is to explore, conceive, design, develop, validate and demonstrate an integrated and collaborative framework combining real and virtual factory data in real time. Through this environment, conceived for any resource, any time and any space, each component can have a holistic view about manufacturing chain information so that their decisions related with the factory and product lifecycles can be taken according to business and operational variations. The dynamic and distributed nature of the new model of manufacturing system based on Industry 4.0 principles can contribute to foster these capabilities. In particular, this study addresses operational and strategic information, automatically captured and continuously published to synchronize the virtual model. As a result, these resources can now report their state in near real-time and use the cloud environments to implement more complex algorithms that can be hardly supported by traditional solutions.

To fully exploit the potential of having all the factory resources connected with each-other, it is essential to explore cross-cutting approaches and solutions that foster semantic interoperability, reducing barriers among data silos. In particular, one major challenge will be identifying valid pattern allowing resources to exchange data among them in an interoperable format. The semantic interoperability of these resources is mainly based on their virtualization. For this reason, following on the need to enhance data integration, a virtualization of factory objects will be essential to abstract complexity of resource-constrained components in virtual counterparts, thus also contributing to facilitate interactions among resources.

Another challenge to be faced will be the exploration of efficient and secure policies to handle the huge amount of data produced within the shop-floor in order to transform these data into relevant knowledge. Indeed, the various types of sensors used to monitor instruments and subjects generate a continuous flow of raw data from which it is possible to extract accurate and updated information, which is then essential to take strategic decisions. For this reason, the environment has to be provided of scalable capabilities that allows to harvest real-time data which can be captured, analyzed and transformed into relevant insights in a secure manner. Under these conditions, a circular data process is generated in which the data flow from physical resources to the back-end (e.g. data store on the cloud), to the analytic systems and then return as control feedback to the physical resources. The efforts have to be addressed in particular towards managing big data, i.e. data sets so complex and large that they can be hardly supported by traditional software tools (McAfee, Brynjolfsson, and Davenport 2012).

In summary, the efforts of the study are addressed to look for an answer to the following research questions:

- *Which strategies and secure policies have to be adopted to empower a synergistic cooperation between real and virtual manufacturing resources during their execution processes?*

- *Which strategies have to be adopted to ensure semantic interoperability among these resources during their cooperation?*

- *Which technologies can support the identified strategies?*

1.4 Motivating scenario

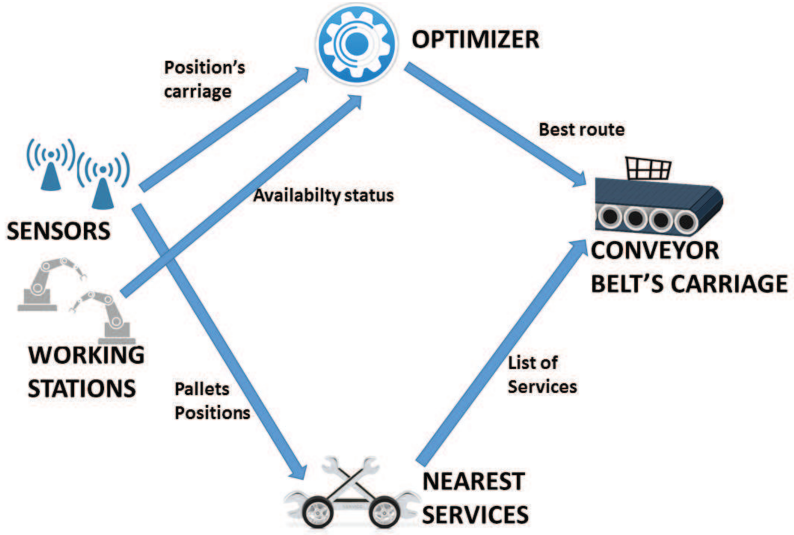
The following scenario highlights the motivations behind this study.

In a factory plant which produces electronic assembly components the production layout is composed by several workstations which can perform different operations (e.g. drilling, milling, etc.). The product components must undergo various operations performed at different working stations and following a specific order. For this reason, they are placed over different pallets to be transported along the production line through a conveyor system. Moreover, the production line is provided of more than one working stations, but current organization of the manufacturing system based on a traditional solution is not able to exploit the eventual availability of some of these machines during the production. Indeed, the scheduling of these machines is typically decided off-line before starting the production, while If adjustments or corrections during production must be applied a manual time-consuming rescheduling has to be done. To make this process more effective and efficient, it is important to automatically move the pallets towards the closest available working station, leveraging the information collected from all the devices concerning their availability status.

In order to achieve this goal, firstly it is essential to monitor and optimize the position of various pallets along the conveyer belt, by enhancing the manufacturing system of the functionalities of indoor localization and route analysis. These capabilities makes it possible to retrieve nearest available services, which are provided by the factory according to the paradigm of the manufacturing servitization. Thus, various kind of devices and services were recently installed across the shop floor. The installed technological components comprise (fig. 1.4): a) sensors monitoring the pallet position; b) Optimizer, a simulation tool which elaborates the pallet position and the availability status of the working stations with the goal to identify the optimized pallet route; c) Smart Actuators, connected to the carriages, which allow to change the route of the pallets along the conveyor belt; d) NearestSer service, providing the list of the nearest available factory' s services, based on the position of the pallet.

The data monitored by the sensors must be sent to the Optimizer, which has to calculate the best route for each pallet. If a pallet route is considered optimizable by the Optimizer, this information must be notified, through a specified alarm, to the actuator linked with the carriage transporting the pallet. Thus, it is desirable a smart mechanism that monitors the pallets routes, captures the eventual events and eventually notifies the alarm towards interested consumers (the actuators), which in turn apply some reactions to the triggered alarms. The above example advocate a kind of synchronization and cooperation among manufacturing components/resources, promoting message oriented middleware and multi-agent development as valid technologies for this kind of solutions. It also clearly emerges that the pro-

Figure 1.4: Overall workflow of the motivating scenario



posed study is challenging for the complexity of the interactions that must be realized among the different resources, as the pattern emerging from the interaction among distributed components is more complex than the ones resulting from their individual behaviors (Leitao et al. 2016). In addition, since the involved production resources can leverage different technological platforms, it is essential to conceive a layer to abstract from vertical communication (integration between MES and shop-floor) and from horizontal communication (integration between different closed-systems). The example also highlights the need to represent the complex shared knowledge in a shared data model which allows to enhance interoperability among the different cooperating resources.

1.5 Contributions

The contribution of this research work is four-fold.

The first contribution consists in an important background knowledge related with the enabling technologies for the realization of the Factories of future. In particular, an overview concerning IIoT, multi-agent systems, message oriented middleware and SWT in the field of manufacturing is reported. This study also includes an extended research to ease cooperation of the various nodes within the IIoT ecosystems. Based on these researches, it is derived the idea of enhancing their semantic interoperability. Main achieved outcomes of this part of the study were disseminated in (Modoni et al. 2017a). The contribution of this work comprises also the exploration of valid strategies for the realization of an interoperable multi-agent system. The concept of virtual collaboration was analyzed by synthesizing the literature which allowed the shift towards knowledge-based environments. Main outcomes of this part of the work were summarized in (Modoni et al. 2017b) and (Kuts et al. 2017). An analysis and synthesis concerning architecture and communication techniques between resources is also studied and completed. It also includes justification for the adoption of SWT, message-oriented middleware and multi-agent systems. At the same time, a qualitative comparative analysis for semantic databases is carried out (Annex 1). Of particular interest is the study for transferring methods and tools (and in particular telemetry technology) from F1 racing cars world to the manufacturing world, based on various identified analogies between the two worlds (Section 3.1). Significant outcomes of the use of the *factory telemetry* were summarized and published in (Modoni, Sacco, and Terkaj 2016).

The second contribution represents the main outcome of this phd thesis, in response to the three research questions posed in section 1.3. According to vision (Section 1.2) and motivations (Section 1.3), this work enables industrial resources to communicate and cooperate among themselves, acquiring, handling and sharing the relevant knowledge about the overall factory. The framework allows to abstract and hide the complexities of hardware or software components involved within the manufacturing system. In particular, it enables a proper interaction of the production resources, enhancing their capability to propagate the information related to changes of state occurring in their context towards other interested resources. For example, with a view to creating the conditions for the intelligent production, a product order can exploit the framework functionalities to autonomously control its moves along the production line, thus opening the door to a more flexible way to manage production.

Another contribution consists in an implementation of an instance based on the proposed framework, in response to the third question posed in section 1.3. In this regard, for each framework's layer the proper technologies are identified and combined in order to realize a software application (*Semantic Event Notifier*) satisfying initial elicited requirements. The realized application allows to demonstrate the feasibility and applicability of the overall conceived approach. Main outcomes of this application were reported in (Modoni, Veniero, and Sacco 2016) and in (Modoni et al. 2017b).

Finally, the fourth contribution consists in an evaluation of the quality of service (QoS) requirements of the *Semantic Event Notifier*, mainly in terms of latency, efficiency, and scalability. This assessment is performed in a Ambient Assisted Living real case study, thus creating the conditions to show the impact of the data storage, of the connected resources and of their requests.

1.6 Structure of the thesis

The thesis is divided into 7 chapters. After this introduction **(Chapter 1)** which examines the vision and motivations behind this work, **Chapter 2** reviews the literature related to this research study. **Chapter 3** presents a system-based analysis to elicit the requirements and constraints for the environment. **Chapter 4** illustrates the framework and its five layers, while **Chapter 5** presents the implementation of an instance of this framework. **Chapter 6** illustrates a quantitative analysis of this implementation mainly in terms of performance within a real Ambient Assisted Living case study. Finally, **Chapter 7** draws the conclusions, summarizing the main outcomes and pointing out some possible future works. Additionally, an annex details a qualitative comparison of semantic database, which embodies one of the back-end of the overall framework.

Chapter 2

Research background and industrial references

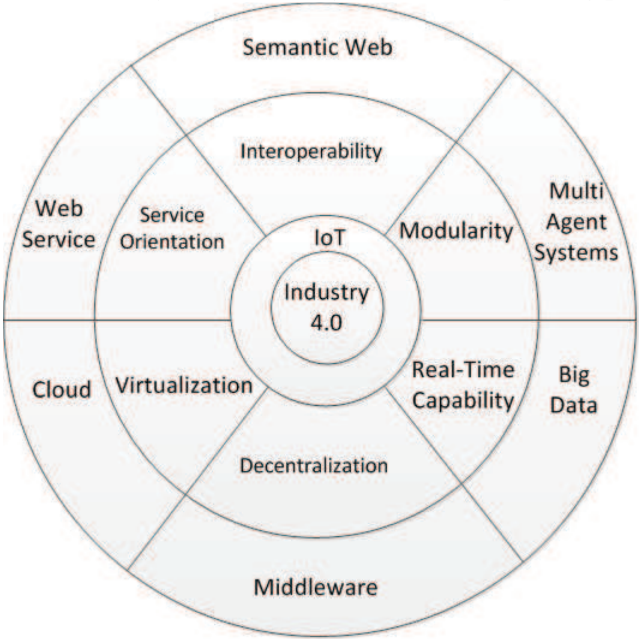
2.1 Introduction

Various areas of the literature are particularly relevant to this research study. This chapter aims to provide an overview on major references in the fields of the scientific and industrial research that have been exploited in this study to provide the foundations for the herein presented approach. The first part of the chapter focuses on main concepts that have been delivered in the area of the Factories of the Future, taking in particular into account the concepts related with the paradigm of Industry 4.0. The second part of the chapter illustrates main enabling technologies for Industry 4.0, focusing mainly on IIoT, SWT and multi-agent technologies whose combination has been exploited to tackle, in this research, the challenges in interoperability and virtualization.

2.2 Industry 4.0

The popularity of "Industry 4.0" paradigm has grown very rapidly in recent years among manufacturers and academics, creating the conditions for a new industrial revolution (Lasi et al. 2014). However, unlike preceding three industrial revolutions which were driven by a single specific technology (e.g. steam, electricity, etc.), this fourth revolution is based on the combination and interaction of a plethora of existing and new technologies around the concepts of IIoT and CPPS (Leitao et al. 2016). The combination of these technologies is realized according with the following six Principles: Interoperability, Virtualization, Decentralization, Real-Time Capability, Modularity, Service Orientation (Hermann, Pentek, and Otto 2016). Fig. 2.1 gives an overall overview of the technologies considered enabling for Industry 4.0 and summarizes their corresponding possible application fields (Uhlmann, Hohwieler, and Geisert 2017). This heterogeneous mix of technologies adds complexity to the digital world of the factory and for this rea-

Figure 2.1: Industry 4.0 Technologies and corresponding principles



son they need to be adapted and rethought in order to address the various needs of complex scenarios such as interoperability and security. Thus, in order to fully unlock the potential of IIoT and CPPS, various existing ICT solutions have to be integrated and adapted to the industrial needs, and then deployed within the shopfloor. In particular, the factories of the future will leverage advances in machine-to-machine communication, wireless sensor technologies, and ubiquitous computing that would allow track and monitor each individual phase of the production.

With the advent of Industry 4.0, several conceptual models or frameworks have been thought in order to clearly highlight the concepts and relationships resulting from the new perspective proposed by the paradigm. Lee et. al. proposed a 5C architecture for Cyber-Physical Systems in Industry 4.0 manufacturing systems (Lee, Bagheri, and Kao 2015). It is intended to provide a step-by-step guideline for developing and deploying a CPS for manufacturing application. The architecture is layers-based and includes the following levels: “Smart connection”, “Data-to-information conversion”, “Cyber”, “Cognition”, “Configuration”.

Another valuable architectural model is the Reference Architecture Model for Industry 4.0 (RAMI 4.0). It combines and integrates the basic elements and properties of industry 4.0 and it is designed to map and classify any component of Industry 4.0 objects and technologies (Hankel 2016). RAMI 4.0 is based on a three dimension coordinate system. The right horizontal axis represents the hierarchy level of a generic object within the factory world; this object can be the manufactured product, a component of a machine, a machine, up to the “Connected World” (thus similar to the automation pyramid, but expanded with the Product item). The left horizontal axis embodies the life cycle of facilities and products, highlighting also the distinction between “types” and “instances” which is crucial to understand the various stages of the lifecycle of any object in the reference architecture. The vertical axis is divided in six layers in order to describe the decomposition of a machine into its properties. The Layers include business process views, functional descriptions, data images, communication behavior including Quality of Service (QoS) as well as Asset Integration through an Integration Layer.

Aiming the interoperability among distributed resources, another key-point is the use of standardized interfaces which ease integration of any (legacy or new) system. In this regard, OPC Unified Architecture (OPC UA) (*OPC UA. Online*) is becoming a de facto standard for enabling a secure, multi-platform and multi-vendor interoperability, thus allowing both the cross vertical cooperation among the ISA 95 layers and the horizontal cooperation among the various IIoT ecosystems. In particular, it offers a Service-Oriented Architecture (SOA) for industrial applications (from factory floor devices to enterprise applications) by exposing an abstract set of services mapped to a concrete technology (Leitner and Mahnke 2006). An interesting aspect of OPC UA is that it defines in its transport layer two different mechanisms for data communication: a TCP protocol for communication at high performance and a protocol based on the Web Services for communication Internet firewall-friendly.

2.3 Interoperability within a manufacturing system

Despite resources involved in a modern manufacturing system adopt a broad range of cutting-edge technologies, they are mostly closed systems with reduced capabilities to cooperate and interact each-other at high-level (Modoni et al. 2017a). This issue is mainly due to a not adequate semantic interoperability of these resources, which represents their capability to automatically interpret the exchanged information in a meaningful manner (Atzori, Iera, and Morabito 2010). Such a condition of a not adequate semantic interoperability, often endemic in distributed systems, can cause the isolation of significant set of data and enlarge the endemic problem of “too much data and not enough knowledge” (Sheth, Henson, and Sahoo 2008). Under these conditions the involved resources can lose, during their interactions, the implicit information about the meaning of exchanged data (Gyrard, Bonnet, and Boudaoud 2014).

The lack of interoperability is mainly due to the fact that developments based on the Industry 4.0 paradigm are characterized by a large heterogeneity in terms of adopted technologies. Moreover, the lack of widely accepted standards contributes to worsen an already messy scenario. Another hurdle is the integration of the heterogeneous models used by each production resources. Indeed, they are typically based on “closed-loop” concepts, which are focused on specific domains and purposes. To contribute to enhance manufacturing resources interoperability, various generic solutions have been proposed in literature. In this regard, a potential approach is defining and representing their shared knowledge through a common model specified leveraging the SWT. For example in (Gyrard, Bonnet, and Boudaoud 2014) it is proposed an approach based on SWT to ease integration among cross-domain applications, through the combination, enrichment and reasoning about Machine to Machine (M2M) communication data. The potential of the semantic-based approach has been also investigated in the context of Virtual Factory Framework (VFF) project which aimed to enhance the semantic interoperability of different software supporting the entire lifecycle of the factory, so that these software can share and exploit the same information (Kádár, Terkaj, and Sacco 2013). In such a scenario, SWT have been adopted to formally describe this information (including the many linking elements) in a semantic model, which provides a holistic view of the factory as a whole, considering resources, processes, product and their coevolution over the time (Tolio et al. 2010). A semantic model is also one of the basis for the implementation of the herein presented framework. Indeed, the idea behind this study is that a semantic model, always synchronized with the real factory, can be a valid basis to implement mechanisms that allow connected resources to easily exchange data with other resources. The design of this model can be based on the reuse of the numerous existing reference models which cover the broad range of the manufacturing knowledge domains.

2.4 Semantic Web Technologies

SWT are increasingly being adopted to model data in a variety of fields such manufacturing, biology, and Public Administration. One of their interesting applications is to support Linked Open Data (LOD) (Bizer, Heath, and Berners-Lee 2009), the paradigm for publishing and connecting structured data on the Web. In the growing landscape of "Polyglot Persistence" where enterprises exploit multiple technologies for data management (Sadalage and Fowler 2012), SWT can play a key role to aggregate and integrate heterogeneous data distributed across many sources. This is due to their aptitude to enhance the semantic interoperability of a technological system, i.e. the capability of the latter to exchange information and exploit the exchanged information (Geraci et al. 1991). Another key advantage of the SWT adoption consists in their capability to support reasoning, which allows to entail and infer new meaningful knowledge about the already defined concepts and their linking relationships. Finally, the SWT can enhance a more efficient semantic search of the information which can play a crucial role to reduce the errors in the search results that are caused by polysemy, synonymy and malformed queries (Modoni, Sacco, and Terkaj 2014a) (Modoni and Tosi 2016).

The Resource Description Framework (RDF) is the W3C standard model for data interchange on the basis of SWT; it provides a general and flexible method to express data as lists of statements in the form of triples composed of subject-predicate-object (*RDF. Online*). One of the RDF strengths is that it allows to express virtually any type of information, without the need to previously adapt its data reference structure as in the relational schema-based approach. As the flip side, the expressivity and flexibility poses the need to revise, for data expressed in such language, some classical data management problems, including efficient storage, indexing and query processing and optimization. In particular, the fine grain of the RDF data models (triples instead of whole records) increases the number of the joins included in the queries, making not trivial the formulation of complex queries and also causing numerous issues of scalability.

The databases for the storage and retrieval of any type of data expressed in RDF are called RDF stores. Their main components are the repository and the Application Programming Interface (API), which communicates with the underlying repository to programmatically expose the main database services. In the past, several stores have been proposed and implemented based on traditional widely tested relational databases. Initially this approach allowed large and powerful solutions to be constructed with little programming effort. However, the flexible model of RDF is poorly suited to traditional relational storage models that, for reasons of efficiency, rely on well-defined structural expectations. Therefore, the more recent trend for managing RDF data has moved away towards so called native RDF stores (also known as Triple stores), i.e. purpose-built databases to handle RDF data, which, not depending on rigid schemas, fits more properly to the flexible structure of RDF data. Thus, they can scale more naturally to large datasets, mitigating some of the performance problems of relational model based stores.

Native RDF store are based on the NoSql databases, whose market is large and not homogeneous. Their most well-known classification groups existing implementations in terms of the following categories (Modoni, Sacco, and Terkaj 2014b): a) Column Databases: (e.g. HBase, Cassandra, etc.); b) Document Databases: (e.g. MongoDB, MarkLogic, etc.); c) Key-value Databases: (e.g. Dynamo, Project Voldemort, etc.); d) Graph Databases: (e.g. Neo4J, AllegroGraph, etc.). This last category comprise the native RDF stores, since RDF data can be thought in terms of a directed labeled graph, where a triple corresponds to a node (subject) - arc (predicate) - node (object) link. However, RDF stores are not “native” graph databases, as they do not support index-free adjacency, nor their engines are optimized for storing property graphs (Robinson, Webber, and Eifrem 2015). In addition to the native triple stores, other solutions (e.g. “native” graph databases or belonging to other NoSql category) are available to handle RDF data, even they are not designed mainly for this purpose (Cudré-Mauroux et al. 2013). Each of these solutions may be suitable and usable for some kinds of tasks and not for others, while a one-size-fits-all killer application for this type of solutions is still not available. Under these conditions, the large number of available solutions to handle RDF data and also the lack of valid benchmarks for their rigorous evaluation make not trivial the task of selection of a valid RDF store during the design of a Semantic infrastructure. In order to choose the most suitable database to be used within a use case scenario, it is important to know the offered features, and related advantages and disadvantages. Since the RDF store is one the backbone of the herein proposed approach, in this study it is proposed a methodological approach that allows to evaluate and rank a selected set of functional and non-functional features of the RDF stores. This qualitative study is reported in Annex 1. In addition, a conceptual map which illustrates main aspects related with RDF stores is reported in fig.2.2.

2.5 Publish-subscribe middleware

A significant contribution to enhance interoperability of IoT nodes can be provided by a middleware application, which is a layer that acts as “software glue” between applications, operating system and network layers (Atzori, Iera, and Morabito 2010). In a middleware, interoperability can be viewed under three different perspectives: network, syntactic, and semantic. While network and syntactic interoperability are well supported by most existing solutions of middleware, many of them lack completely support for semantic interoperability (Razzaque et al. 2016). Indeed, as reported in Section 2.3, in IoT scenario semantic interoperability is very challenging for the heterogeneity of devices and the lack of standards. A middleware offering semantics capabilities, though limited, is LinkSmart, which is the main outcome of the Hydra project (Kostelnik, Sarnovsk, and Furdik 2011). This solution combines SWT with SOA principles to provide a mechanism for wrapping API interfaces of various physical devices with a specific web service extension, which is enriched by a semantic description of generated WSDL files. In this way, it allows to generate a stub within the related client code according to the device’s capabilities specified in a semantic model. An evolution of the LinkSmart middleware is reported in (Patti et al. 2016), where it is presented the design and implementation of an event and service-oriented middleware based on SWT for energy efficiency in public and private spaces. In such a solution, application developers can query different kind of information from a semantic model which can be accessed through a SPARQL end point. The semantic model comprises the information about location and capabilities of a sensor but it also includes the complete list of sensors distributed in a room or the list of actuators provided of a specific control capability. However, in such a solution, it is not addressed the issue of synchronizing real and virtual world, which is a secondary goal of the herein presented research.

Among the several types of available middleware, an efficient solution to connect and integrate real and virtual resources is offered by the technologies based on the publish-subscribe interactions. In fact, they well adapt to the needs to synchronize the changing information among decoupled components distributed within an IoT platform, thus fully exploiting the potential of the IoT (Ali et al. 2015). However, they lack mechanisms to express both the data requests from the resources and the notifications events in a flexible and expressive way, obligating the subscriber to know the topics offered by the publishers and to be able to process natively the published messages (Modoni et al. 2017b). Indeed, to the best of the knowledge, solutions currently available in literature support only static mechanisms of selection of the data to be exchanged, which are based on predefined syntactical subjects

Figure 2.2: A conceptual map which illustrates main aspects related with RDF stores

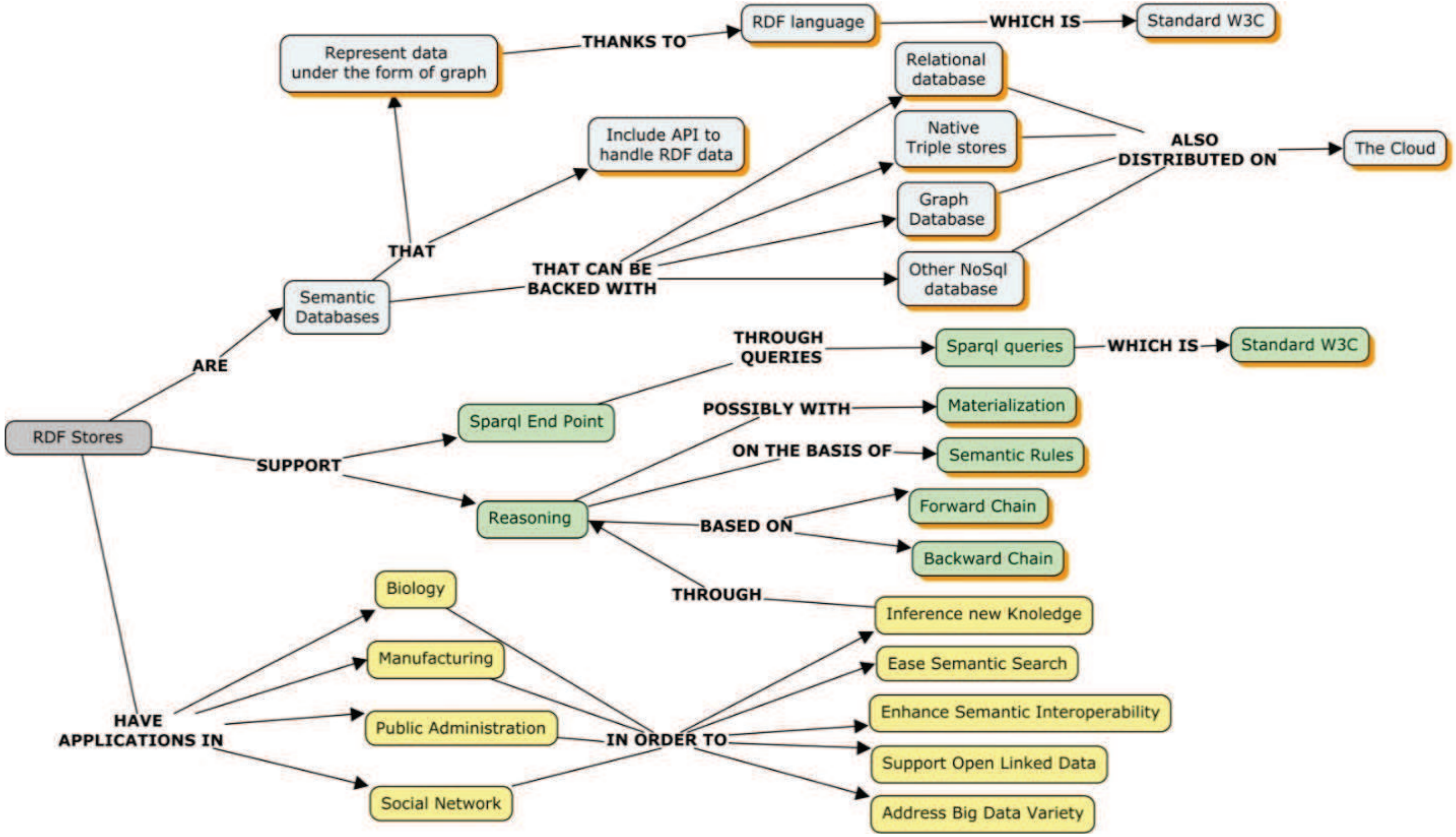
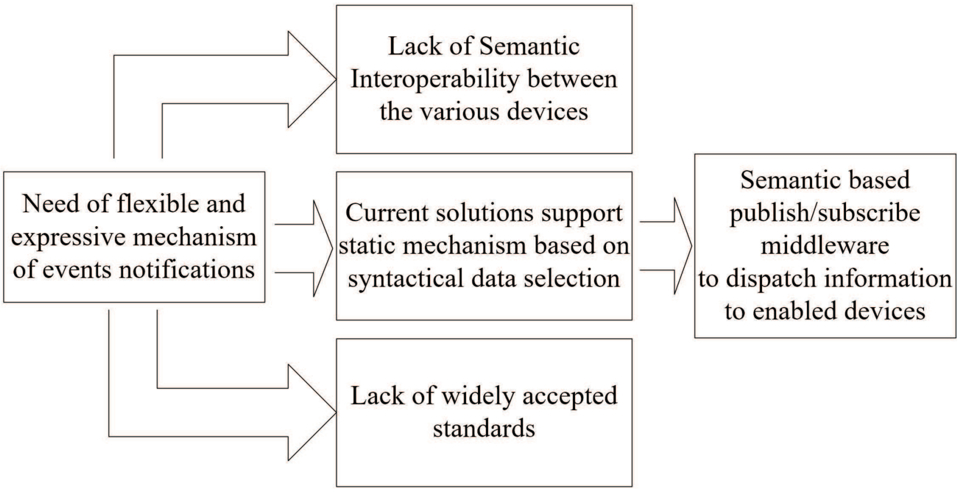


Figure 2.3: Definition of the research problem referring to the exploration of expressive mechanisms for events notification



(Fortino et al. 2013), while they do not consider different expressivity requirements normally needed in the definition of complex scenarios such in the manufacturing domain. The semantic enrichment of the data allows to overcome this current limitation, mediating between the consumers requests (expressed according to a specific data model) and the updates produced by the publishers (expressed according to the same data model), thus achieving a more flexible and expressive approach compared to syntactical ones (Moser et al. 2009). This research problem, summarized in fig. 2.3, will be studied in deep in the following chapters.

2.6 Agent-based systems

In order to enhance agility and flexibility of an IIoT based application within the shop floor, a possible way is conceiving it as a distributed cooperative system based on agents. In such a context, behind each production resource there is a specific agent which manages its collaborations. The agent proxies the corresponding resource or one its function and then can cooperate with other agents to proactively collect data and update the current state of the system. Multi-agent based technology has been applied in different fields (e.g. telecommunications, healthcare, manufacturing, etc.). On agent technologies it is inspired the architecture proposed by the research project UBIWARE (Katasonov et al. 2008), which realizes a middleware for the IoT to support the creation of a self-managed system comprising a set of distributed and heterogeneous components of different nature. Through this approach, the information related to each resource's condition and their interactions are monitored and then persisted into a storage. In the aerospace industry, NASA also developed an agent-based solution to balance the load of multiple demands over its satellites (Luck, 2005).

The holonic manufacturing can be considered one of the first agent-based solution proposed in the manufacturing field (Van Brussel et al. 1998). Holonic manufacturing is represented by holarchies of holons, which consist in autonomous, cooperative entities that can represent any manufacturing resource. A holon, uniquely identified within the system, may be composed of different subparts and at the same time it can be part of another larger part. Also, DaimlerChrysler used multi-agent technology to support its autonomic computing systems aimed to increase their productivity (Leitao et al. 2016).

More recently, an agent-based system is introduced in (Wang et al. 2016) where it is proposed an intelligent negotiation mechanism for agents which can cooperate with each other. Furthermore, the study illustrates that complementary strategies can be designed to prevent deadlocks by improving the agents' decision making and the coordinator's behavior. In (Rocha et al. 2014) it is illustrated another agent based framework for manufacturing systems which provides the latter the capacity to quickly adapt and reconfigure, allowing to cover the plugging and unplugging of production resources as well as the adjustment of parameters during production activities.

Despite the advantages offered by multi-agent technology in terms of reconfigurability and flexibility, the adoption of this technology in real scenarios is limited for two reasons: 1) the cost for implementation is often higher compared to traditional control strategies; 2) the difficulty of various devices to run the agents (Leitão 2009). For these reasons it is reasonable to move a multi-agent based application towards a micro-services architecture which can facilitate the orchestration of the legacy devices. Besides the issues related to integration of legacy devices, in the communication among distributed and autonomous agents it is important to preserve the semantic content of exchanged messages. These distributed entities need to have a shared understanding of the concepts related with their domain knowledge. This understanding can be given under the form of semantic model. In this regards, an interesting combination of semantic model and multi-agent technology in within the ADACOR architecture, which is based on the Holonic Manufacturing Systems paradigm. ADACOR is developed as a set of autonomous, cooperative holons, each of them represents a specific manufacturing component. The latter can be a physical resource (e.g. numerical control machines, industrial robots, etc.) or a logic entity (e.g. orders, etc.) (Leitão and Restivo 2006).

Implementation of such semantic models are already in use in multi-agent based solutions like the Java Agent DEvelopement (JADE) framework (Bellifemine, Caire, and Greenwood 2007). However, this framework uses a proprietary semantic model which relies on the Java inheritance. While this approach is comfortable (especially in conjunction with ad-hoc Java classes which allow serialization and deserialization process to object level), it's hard to integrate the framework in a flawless flow of information because of the non-trivial translation between standardized ontologies like OWL and JADE proprietary format. In addition, JADE uses only semantics for the provided vocabulary and the syntax checking, while the adoption of a semantic model can bring the significant advantage of creating a cognitive model with reasoning capacities (Lemaignan et al. 2006). The exchange of semantic messages in JADE are based on a restrict meta-model which comprises entities such as Concept, Agent Action and Predicate. Due to the fact that these messages are FIPA compliant (*FIPA. IEEE Computer Society. Online*), JADE agents can interact with other agents, which can be also agents managed from other FIPA frameworks different from JADE. The combination of semantics with JADE can be also found within the project GRACE. In such a context, the asynchronous communication among distributed agents included also the content meaning of the exchanged messages, which were standardized according to the GRACE ontology (Leitão et al. 2015).

2.7 Middleware's Quality of Services

To guarantee that applications meet the requested Quality of Service (QoS) requirements in terms of performance, scalability and security, it is essential that the platforms on which they are built are tested and evaluated using specific benchmarks (Corsaro et al. 2006). However, if a benchmark is to be useful and reliable, it must fulfill several fundamental requirements. First, it must be designed to stress platforms in a manner representative of real-world messaging applications. It must exercise all critical services provided by platforms and must provide a level playing field for performance comparisons.

Different metrics can be used to facilitate a standard and systematic performance evaluation of a semantic application when dealing with huge flows of near real-time information coming from many sources; two of the most commonly used metrics are query duration and load time (Modoni, Sacco, and Terkaj 2014b). Other useful metrics include disk space requirements, memory footprint and deletion duration. Based on the above metrics, several benchmarks (e.g. LUBM (Guo, Pan, and Heflin 2005), Berlin Benchmark (Bizer and Schultz 2009), etc.) have been formalized and published. Nevertheless, all of them focus on the ontology related performances, but none of them referring to the emergence of information and to their dispatching in an IoT centered application. For this reason a specific testing and more suitable benchmark framework will be designed in this research work to assess QoS's applications based on the proposed framework.

Chapter 3

A system-based analysis

3.1 Towards a factory telemetry

Analogies can be a valid way of analyzing the performance of industrial processes in order to understand potential improvements. A significant example is represented by the several parallels between biological and manufacturing systems that have been drawn in literature to solve a series of problems of the modern manufacturing, through the study of the structure, control mechanisms, and functions of the biological systems (AlGeddawy and ElMaraghy 2010). The idea behind our vision is that various analogies can also be observed between the worlds of F1 racing cars and modern factories (Modoni, Sacco, and Terkaj 2016). In fact, like the F1 cars, a manufacturing environment comprises a set of processes to be monitored in near real-time, huge information flows (and corresponding software applications) from which to take critical decisions in limited time (*Factory Telemetry*), and a team of people that has the task of developing, maintaining, measuring, and adjusting the system under changing conditions. Moreover, it must be emphasized that F1 represents a relevant reference case, since it is always on the cutting edge of technological development.

3.1.1 Similarities with the racing car

Five main similarities between these two worlds are identified (fig. 3.1). They are reported in the following section while they were deep in (Modoni, Sacco, and Terkaj 2016).

Accurate monitoring of the assets for critical decisions making
Telemetry is a proven technology of F1 through which a deluge of data is transmitted from the car to the pitwall in order to allow a team of engineers to monitor accurately and constantly several parameters about car systems such as suspensions, engine, transmission, and wheels (Cocco and Daponte 2008). In this way the engineers can watch over the racing car performance and optimize the vehicle setup, suggesting drivers to change one of these parameters. Moreover,

they can use the telemetry to analyze tactics and strategies, investigating on which corners car could go faster. The accurate monitoring supported by a similar factory telemetry would be relevant for any manufacturing company where data provides the basis for critical decisions making. In particular, there are two major areas of associated benefit: the management of the allocated resources, and the continuous improvement between design, development, and manufacture of the products (enabling a kind of loop between the three stages). Along the whole factory life cycle, the sensors connected with the real factory components can provide detailed information about the performance of various processes, ensuring a better visibility and control of the used resources and a more reliable forecasting. Moreover, a proper integration of the data coming from telemetry and from enterprise systems such as MES or ERP could help operations managers to analyze the dynamics of the manufacturing processes and seek to identify potential improvement actions (e.g. reconfigurations of the input parameters, changes in the management of maintenance activities, etc.). In this way it is also possible to identify and address any bottleneck and ensure a smart utilization of expensive machineries which allows to maximize the throughput. Finally, the analysis of the gathered data enables also the check if the product "as built" is compliant with the specifications and requirements of the designers, helping the company to adjust and optimize processes between design and production stages. Specifically by merging the designer specifications about how the product is to be manufactured and the information about how the product is actually being manufactured, it is possible to build an instantaneous perspective on how the manufactured product is meeting its design specification goals.

Feedback from virtual to real to apply corrective decisions

The F1 two-way telemetry is a bidirectional data flow that allows engineers to make real time adjustments remotely on the car even while the latter is running on the track. In this way it is possible to align the setup of the car with the needs of the driver also taking into account external conditions. From the 2003 season, the two-way telemetry has been banned from the FIA (Federation Internationale de l'Automobile), with the exception of the system for the activation the DRS (Drag Reduction System), which allows the driver to adjust the rear wing in order to reduce drag and increase top speed. In fact, this system is automatically enabled only in certain circumstances on the basis of the data coming from the cars telemetry (FIA 2011). Similarly, within the factory, a two-way telemetry would allow project managers and designers to accurately monitor manufacturing processes progress in real time, enabling them at the same time to detect problems early (e.g. breakdowns) and apply corrective decisions based on the information they receive and analyze. Once these decisions are final, they would be applied to the real factory, thus implementing the closed loop between the virtual and real factory (Kádár, Terkaj, and Sacco 2013).

Integration with Advanced Simulation and Forecasting

Telemetry not only allows F1 teams to collect and monitor information in real time but also to use them in order to properly simulate the car for maximizing its performance. These simulation models have become so advanced that potential lap time of the car can be calculated, and this time is what the driver is expected to meet. Moreover, between a race and another, the F1 teams compute a series of analysis through which they are able to build predictive models of how the car will perform with different setups, different tracks under changing ambient conditions, on the basis of the collected historical data (Waldo 2005). If a telemetry-based simulation is used on the factory floor next to the machineries that it models, it could give operators a digital representation that looks and acts exactly like the machine itself. In this way it can offer the capability to execute the operations through a simulation environment where the various product components can be inserted and tested in different configurations across the entire production chain. Under these conditions, operators can optimize and validate new processes into state-of-the-art machine, without taking the latter out of production. In order to realize this approach, the data telemetry should be fully integrated with discrete or continuous simulators, which allow to model the complex dynamics of a manufacturing system. The latter can refer to the processes of a single cell, a production line, an entire factory, or several companies interconnected with the warehouses through a network. Another key success factor of the approach is the capability to initialize the simulation models through a snapshot of the real system (Kádár et al. 2010).

Digital continuity between telemetry historical data

The simulation-based analysis of the F1 car performance mentioned in the previous subsection can be effectively exploited only if the digital continuity between telemetry historical data is guaranteed. Indeed, it must be ensured that data can be playbacked and passed as input to the simulation tools in order to perform forecasts against which to compare the behavior of actual running real-time systems. Digital continuity is also important from a reliability point of view, since statistics based on historical data make sure that installed components not exceed their recommended lifetime ranges (Waldo 2005). Finally, digital continuity plays an essential role in case of an accident, since FIA can determine driver errors as a possible cause on the basis of the driver inputs that have been recorded. Similarly, digital continuity between historical data of factory telemetry allows to create numerous simulated data streams that are semantically interoperable with real operational data. Such data emulation offers a real-world environment to train personnel, where for example control room operators can directly interact with the system and receive real feedback (Capozzi et al. 2014). Specific analytics have to be performed on the gathered information to extract better insight over the progress and status of each single machine. These analytics can provide comparison between machine performance. Moreover, historical information can be measured to predict the future behavior of the allocated machineries. In order to guarantee the digital continuity between historical data, Terkaj et al.

proposed to use an history model of factory objects (Terkaj, Tolio, and Urgo 2015). In this way, historical data can be collected and stored in a distributed way, while keeping an overall coherence thanks to a common virtual factory model.

In situ simulations
The seamless integration of simulation tools and the real environment of the factory paves the way to in situ simulation approaches, which takes place in the working environment and involving those who work there. The in situ simulation is distinct from center-based simulation, which is performed in a context separated from the work environment (Terkaj, Tolio, and Urgo 2015). A similar philosophy can be found in F1 behind the driving simulator, which is a car cockpit that gives drivers true feel of a real environment and direct feedback on their actions. The driving simulator replicates real race track conditions and is used to test different aspects that affect performance of the car such as wings and brake settings. The high fidelity of the simulator allows the driver to feel the difference that modifications applied to the car setup can produce without the high acceleration of a real test drive. As the new FIA regulations limits the number of test days on the track and also wind tunnel time to reduce costs and level the playing field, the driving simulator plays a key role for drivers training, saving at the same time both time and money while respecting new regulations. Moreover, the driving simulator can be used to test future car designs and train new drivers on different circuits.

3.1.2 A dedicated software for telemetry analysis

On the basis of the analogies reported in the previous section, it is interesting to experiment a transfer of methods and tools from one field to the other. Specifically, the focus is oriented on a set of relevant features of the F1 telemetry (fig. 3.2). The aim is combining and adapting methods and techniques borrowed from the F1 telemetry to explore valid solutions to support the Factory Telemetry. The exploitation of the factory telemetry could offer various methods to perform valuable simulations of the production processes, using as input the data coming from the real factory.

Our study focused in particular on the requirements elicitation of a software application that receives, interprets, persists, integrates, and analyzes the collected data of the telemetry stream. During this activity, a valid starting point can be the evaluation of existing F1 telemetry systems, such as Atlas (*Atlas. Online*), which is the standard system, or Wintax (*WINTAX. Online*).

The following list highlights the major features that a software application supporting factory telemetry should provide: capability to maintain the links between factory configurations/layouts and telemetry data; simulation of the effects of different input parameter values on a given factory process; and direct comparison of simulated results with real telemetry data or with other simulations. Data visualization is an essential task of the envisioned software tool. XY Charts, waveform and scattered plotting, statistics and animations permit to show under different views the data telemetry acquired from the sensors which are connected to the real factory. In this way, it is possible to study accurately a particular aspect of the factory. Among the most significant graphical features, the new environment should include functionalities to filter and select a part of the collected data stream in order to provide it as input of a new simulation. Using a multiscale model as reference, the envisioned software application should also comprise capabilities to zoom in and zoom out the selected data in order to drill down into specific data subsets. Moreover, the Graphical User Interface should also provide facilities to change the factory setup which comprises the different input parameter values for the proper configuration of the factory processes; the setup can be stored to a database in order to be used as input for a following simulation. Each new created setup should be compliant with the previous already saved setups, allowing in this way to guarantee their Digital Continuity.

A typical issue of the data coming from sensors is the noise errors. As it is better to have a smooth curve to analyze the factory performance, removing high frequency noise and spike is a necessary feature for the envisioned software application. In this regard it is essential to use various techniques of high frequency noise removal such filtering and smoothing. Also, the end-users should have the possibility to introduce a sensor offset/gain or implement a sensor correction. Combining the digital versions of telemetry signals and a lot of math/logical/filter/statistical functions, also through the integration with external commercial tools such as Excel, Matlab and Simulink, it is possible to create the so-called virtual channels, which represent a method to abstract and remap the original telemetry channels (for example to create alarms). A proper API (Application Programming Interface) should guarantee the access to telemetry data, enabling data analysis in external tools (e.g. Matlab). Finally, a Multicast transmission of the data over the factory network would allow the software application to receive the telemetry regardless of the PC where the software application runs, as long as it is connected to the network and enabled.

3.2 Analysis through System dynamics model of the Middleware

A major goal of this research is a system capable of propagating contextual state messages coming from physical and virtual sensors (data publishers) to keep the other interested components (data consumers) updated about interesting events. Notification are distributed as messages to other devices (consumers), so that the latter can perform further evaluations and actions, where needed. Moreover, consumers should be able to subscribe to events of interest simply specifying the kind of information they are interested in without bounding to specific knowledge about the producers' capabilities.

The growing quantity of devices and services that are connected to the Web

Figure 3.1: Similarities between the worlds of F1 racing cars and modern factories

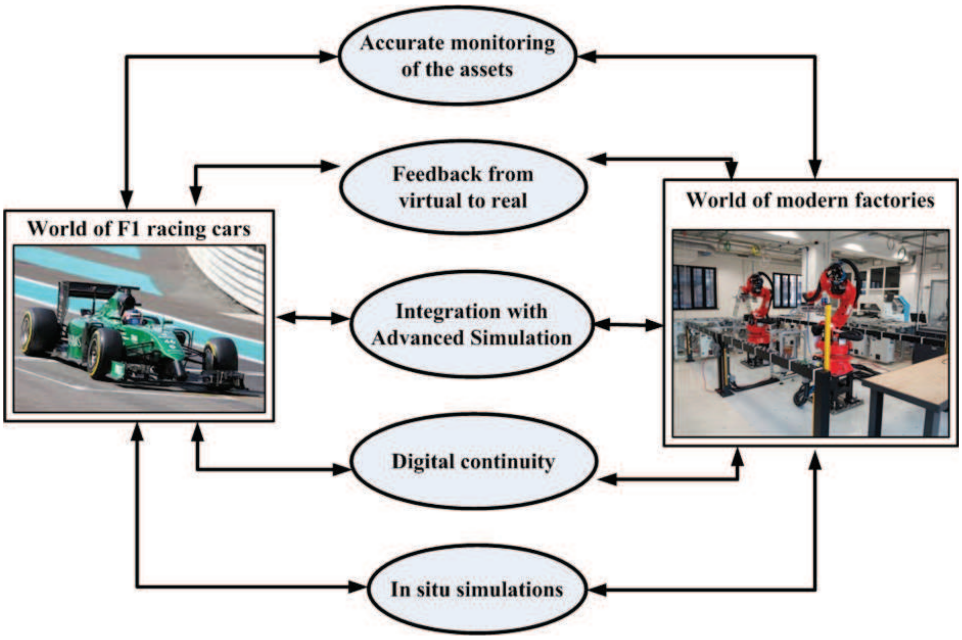
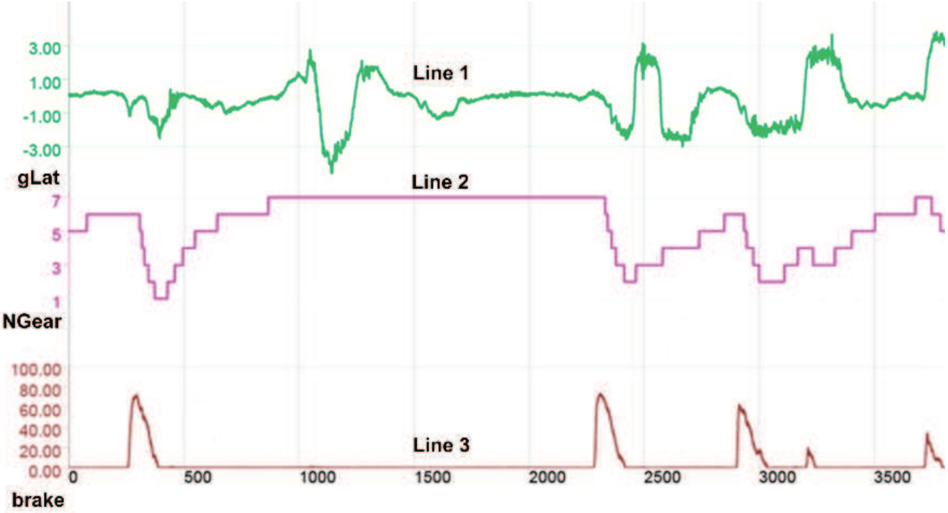


Figure 3.2: Telemetry of a F1 car which contains speed, gear and other channels (SOURCE: Caterham F1 Team/Renault Sport F1)



makes it more difficult to deal with the communication issues among the several actors of IoT system (devices, services, and so on). Thus, the system must be supported by a scalable architecture, that reduces memory and computational cost of a massive amount of real-time data produced by devices. Moreover, the implementation of a Trust model enables reliable and secure interactions between trustworthy entities. In this regard, the solution of Trust model is based on the evaluation of the devices reputation, so that only the devices that are granted can publish critical information while the other are stopped.

The analysis presented in this section aims to answer the following questions:
a) Which are the variables influencing the success of the overall system?
b) Which are the (technological) dimensions affecting the performance of this system?

Due to complex characteristics behind the above described scenario, the simulation of the system's dynamics can be considered a systematic work that supports the study of the relations linking the key-factors affecting the success of the system. In this regard, it can be used the System Dynamics (SD), a methodology with a systematic approach which can help to obtain insights into problems of relevant complexity figuring out the relationships among the variables (Forrester 1995).

Following the SD methodology, table 3.1 reports a list of all the identified parameters linked with the success of the various stages of the system's design, development and operation. For each parameter it is reported the description, type (stock, flow and auxiliary) and units. Leveraging the identified parameters, the causal-loop diagram is constructed (fig. 3.3). Within this diagram, the cause and effect relationships among the parameters reported in table 3.1 are identified. There are four causal loops in total where two are reinforcing and the other two are balancing. Fig. 3.4 represents Stock Flow Diagram of the subsystems handling the middleware requests.

Moreover, we realized some experiments by changing some of the auxiliaries defined in the SD model, using the tool Powersim[1]. In particular, we did the simulation of the Publishers stock over time (fig. 3.5) as the integration of the inflow "enabled publishers" and outflow "blocked publishers". In this representation, the two flows are simulated by changing their values over time (x axis) (fig. 3.6).

Table 3.1: List of key-factors the IoT infrastructure

VARIABLE	UNITS	TYPES	RANGES
Rate of enabling Publishers	Number of enabling Publishers per minute	Flow	>= 0

---

[1] http://www.powersim.com/

Publishers	Number of enabled Publishers	Stock	0..1000
Rate of blocked publishers	Number of blocked Publishers per minute	Flow	>= 0
Blocked Publishers	Number of blocked Publishers	Stock	0..1000
Rate of intrusions	Number of intrusions per minute	Flow	>= 0
Disclosed intrusions	Number of Disclosed intrusions	Stock	>= 0
Average Intrusions per Volume	Number	Auxiliary	>= 0
Number of Kbytes per intrusion	Kbytes	Constant	1000
Rate of compromizes	Kbytes per minute	Flow	>= 0
Number of compromised data	Kbytes	Stock	>= 0
Measures for security	Number of encrypted connections	Auxiliary	>= 0
Max threshold compromised data per publisher	Kbytes	Constant	3000
Gap compromized data / publisher	Kbytes	Auxiliary	>= 0
Number of Changes per minute	Number	Constant	60
Number of Kbytes per change	Number	Constant	10
Rate of Changes applied to DB	Kbytes / minute	Flow	0..300 Mega per second
Rate of enabling consumers	Number of enabling consumers per minute	Flow	>= 0
Consumers	Number of enabled Consumers	Stock	0..1000
Rate of disabling consumers	Number of disabling consumers per minute	Flow	>= 0
Gap between current average request time and its threshold	Milliseconds	Auxiliary	

Data volume	Kbyte	Stock	>= 0
Measures to increase security	Number of encrypted connections		
Data quality	Number of compromised data (Kbyte)	Stock	>= 0
Trust of consumers	Percentage of trust of a consumer towards a publisher		0..100
Number of requests per consumer	Number	Constant	5
Sent requests	Total number of sent requests	Stock	>= 0
Rate of sent requests	Number per minute	Flow	>= 0
Pending requests	Total number of requests waiting an answer	Stock	0..500
Requests acceptance rate	Number per minute	Flow	>= 0
Accepted requests	Total number of accepted requests	Stock	>=0
Consume of computational power of server	Kbytes of CPU Consume	Auxiliary	> 0
Average Time of the requests	Milliseconds	Auxiliary	
Maximum threshold for the time of the request	Milliseconds	Constant	500
Nodes of the server	Number of enabled node of the server	Constant	>= 1

Figure 3.3: Causal loop

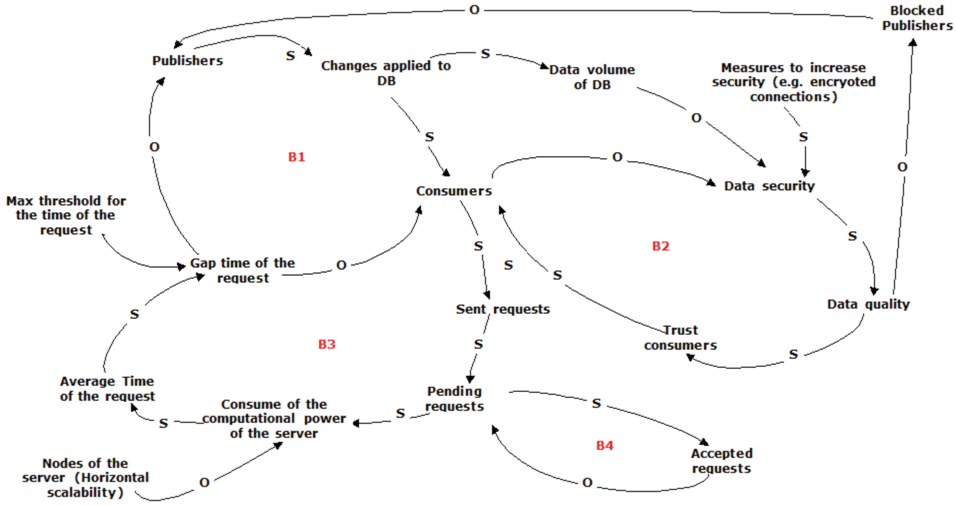


Figure 3.4: Stock & Flow Diagram of the subsystems handling the middleware requests

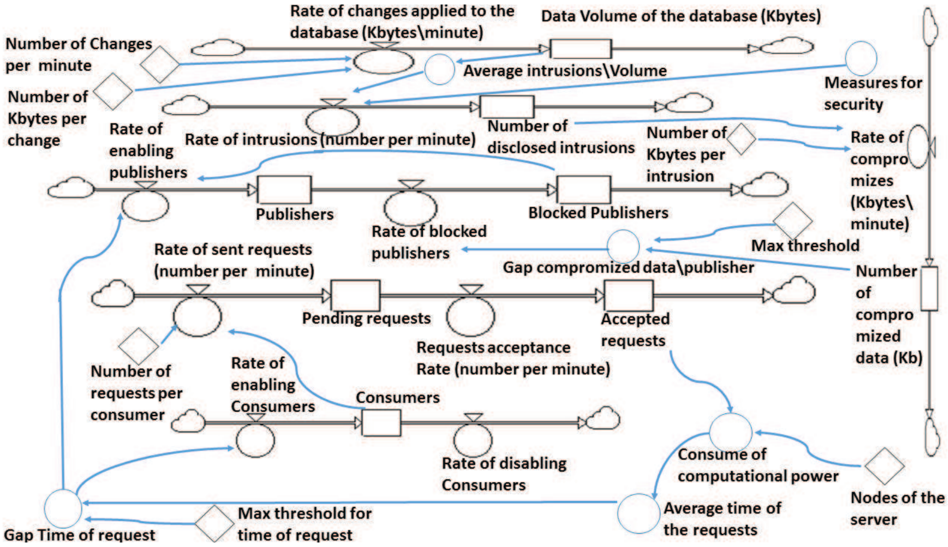


Figure 3.5: Simulation of the Publishers stock over time

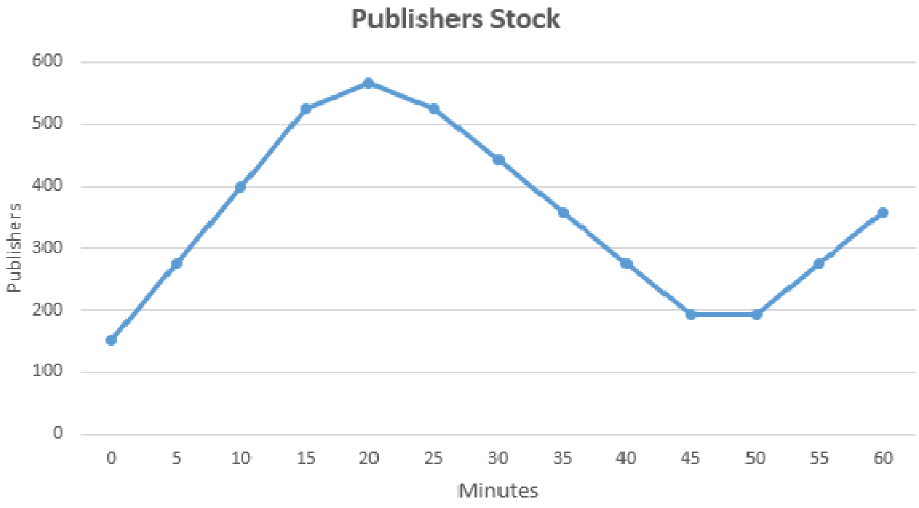
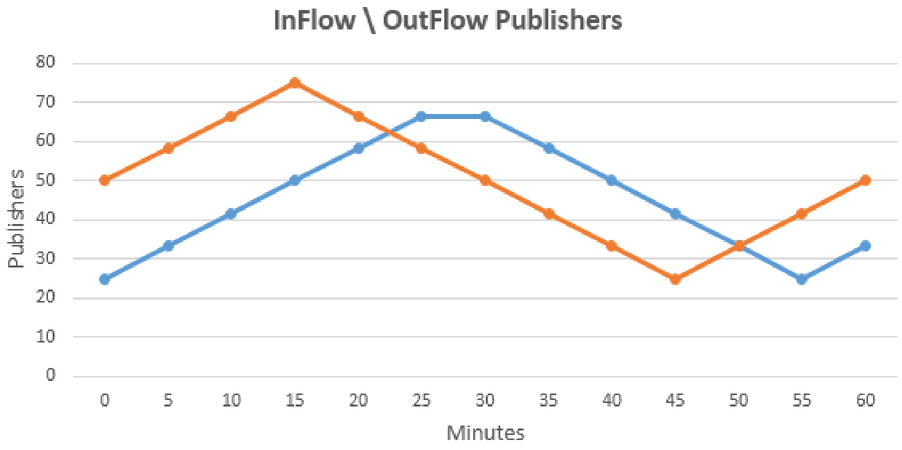


Figure 3.6: Representation of the inflow “enabled publishers” and outflow “blocked publishers” over time (x axis)



3.3 Strengthening the cyber-security of the manufacturing systems: a methodology compliant with the NIST Cyber-Security Framework

3.3.1 Introduction

Cybersecurity is one of the most challenging topics in implementing the paradigm of Industrie 4.0. While manufacturing companies are striving to exploit the opportunities offered by new advances in ICT, they have to face an increasing number of threats and vulnerabilities, which can jeopardize the cyber-security of the various deployed information systems. In particular, data confidentiality, integrity, and availability (the CIA triad) are the main features of these systems that can be compromised. Hence, it is essential to identify new security measures that mitigate the risk of internal and external attacks against the systems to an acceptable level, thus protecting the information processed by those systems (Wells et al. 2014). In this regard, several well-known international organizations (e.g. ISO, SANS, and NIST) conducted various cybersecurity initiatives. Among the most relevant ones, the National Institute of Standards and Technology (NIST) released the "Framework for Improving Critical Infrastructure Cybersecurity" (CSF) (*Cybersecurity Framework Manufacturing Profile. Online*) in order to provide a guidance to assess and manage cybersecurity risks. The NIST framework enables an organization - regardless of sector, size, degree of risk, or cybersecurity sophistication- to apply effective practices of cyber-risk management in order to improve the security and resilience of its digital infrastructure. Baldoni et al. tailored the CSF to the needs of the italian SMEs (Baldoni and Montanari 2016), while in (*Cybersecurity Framework Manufacturing Profile. Online*) NIST provided the Cybersecurity Framework implementation details developed for the manufacturing environment, focusing in particular on five common business/mission objectives: Maintain Human Safety, Maintain Environmental Safety, Maintain Quality of Product, Maintain Production Goals, and Maintain Trade Secrets. A manufacturing company can successfully exploit the potential of a framework such as the CSF only if its application is framed within the business context in which the organization operates. The current state of the art shows that cyber-security models are typically decoupled and separated (isolated) from the business processes models, and this lack of integration fragments the efforts of the organizations addressed to strengthen the cyber-security. The main objective of the work described in this section is to analyze and assess the manufacturing systems cyber-security aspects integrated into the context of the organization's business model. In particular, we introduce a methodology to align and correlate core activities provided by the CSF with the so called factory coevolution, a model which represents the dynamics behind a manufacturing company and which focuses in particular on the three main entities that characterize the whole lifecycle factory (i.e. products, processes, production systems) (Tolio et al. 2010). We firmly believe that such a process of alignment could greatly help organization's analysts to identify and manage cybersecurity activities priorities, allowing to allocate investments towards specific compliance objectives, thus reducing the overlapping of existing resources.

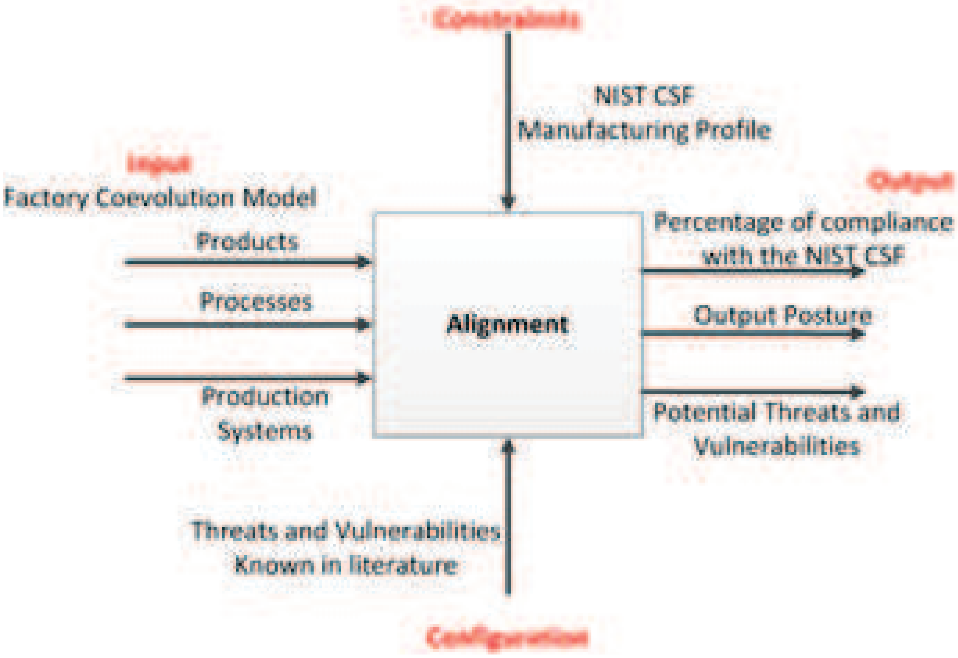
3.3.2 The Alignment process

The definition and workflow of the Alignment process is given in fig. 3.7 by using an IDEF0 model (*IDEF. Online*), which represents the process as a box, the inputs as horizontal arrows entering the box and the outputs as horizontal arrows exiting the box. The model represents also constraints and configurations of the process under the form of vertical arrows, respectively entering the box from the top and from the bottom. The input of the process is the current representation of the factory's world comprising mainly a list of their instances of products, processes, and productions systems, their related attributes and their linking relations. The constraints of the process consists in the manufacturing profile of the CSF (*Cybersecurity Framework Manufacturing Profile. Online*), represented under the form of a list of conditions that company has to meet in order to be compliant with the CSF, while the configuration consists in a list of threats and vulnerabilities known in literature. The output of the activity "Alignment" is the cyber-security posture of the company which refers to the specific factory coevolution given as input. The resulting output posture can then be compared with the target posture. This comparison allows to highlight an eventual existing gap of the organization to reach the compliance with the CSF, measuring in particular this gap under the form of percentage of compliance. The eventual existing gap should be addressed by the specific organization in order to meet its cybersecurity risk management goals, also through an action plan which allows to prioritize the mitigation measures. In particular, an iterative and continuous process of manipulation and balance of internal and external forces impacting on the factory coevolution can help to maximize the compliance with the CSF, also according with the business process management plan. These changes can be applied to the active processes (e.g. introducing new technologies), to the products (e.g. using new material or implementing new features) or to production systems (e.g. updating the production planning and control). The last output of the activity "Alignment" is a list of potential threats and vulnerabilities provided on the basis of the mapping between the input factory coevolution and the list of the threats and vulnerabilities known in literature.

3.3.3 The conceptual model

The complexity and heterogeneity of the information involved in the alignment activity requires the availability of an high-level conceptual model, able to capture the concepts and their corresponding relationships. For expressing security aspects in business process models, the literature offers various modeling languages. In particular, the graphical modelling languages (e.g. SecureBPMN (Brucker et al. 2012), UMLsec (Jürjens 2005)) are easy to learn and to use, but they are limited

Figure 3.7: IDEF0 diagram for the alignment activity



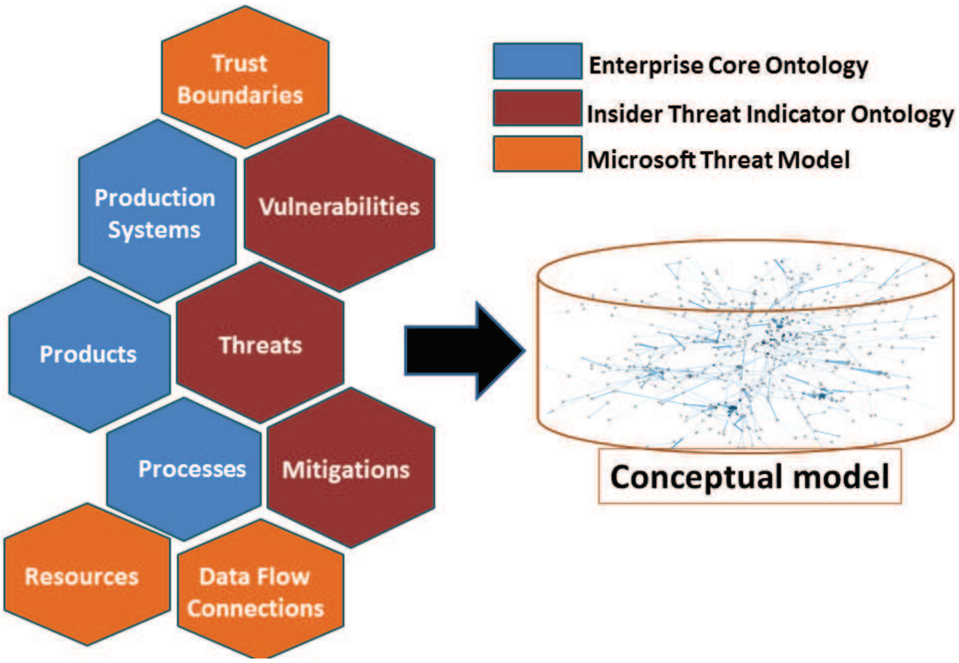
in terms of expressiveness and thus they do not permit, for example, to define ad-hoc security policies. Another potential approach for expressing security aspects is based on SWT languages, which are more expressive and flexible, enhancing also the definition of rules which offer adequate support to check the compliance with the framework. An important challenge to face along the present line of work is the implementation of the conceptual model from scratch, which is extremely expensive if the application domain is complex, such as in industrial use cases. Nowadays, a large number of very comprehensive reference models are available, covering a wide range of domains; reusing them can be related to significant costs, which greatly reduces the costs of a new implementation. In this regard, a fruitful starting point for the implementation of the needed conceptual models is the reuse of existing reference models already present in the literature (Fig. 3.8). In particular we leverage the following reference models:

- **the Enterprise Core Ontology** (Modoni et al. 2017a) (Modoni et al. 2015), that digitally represents the world of the factory;

- **the Insider Threat Indicator Ontology** (Costa et al. 2014) which is a formalization of insider threat detection, prevention, and mitigation;

- **the Microsoft Threat Model** (Shostack 2014), which represents the conceptual model on the basis of the Microsoft Threat Modeling Tool 2016.

Figure 3.8: Reuse of existing reference models to implement the conceptual model



The result of the reuse process is an abstract conceptual model which reconciles, links and integrates various models coming from different domains. Fig. 3.9 and fig. 3.10 show the excerpts of two of these models, under the form of UML class diagram. Fig. 3.9 reports the model corresponding to the ID-AM category of the CSF which expects that each technological (hardware and software) component is administered by a responsible (highlighted in red to show the high priority), is inventoried and documented (low priority) and each its use is traced through automatic mechanisms (moderate priority). Fig. 3.10 reports an excerpt of the model for representing the technological resources, their linking connections and the threats that jeopardize their cyber-security. The mapping between these two models let to highlight their potential overlappings.

Figure 3.9: The model corresponding to the ID-AM category of the NIST CSF

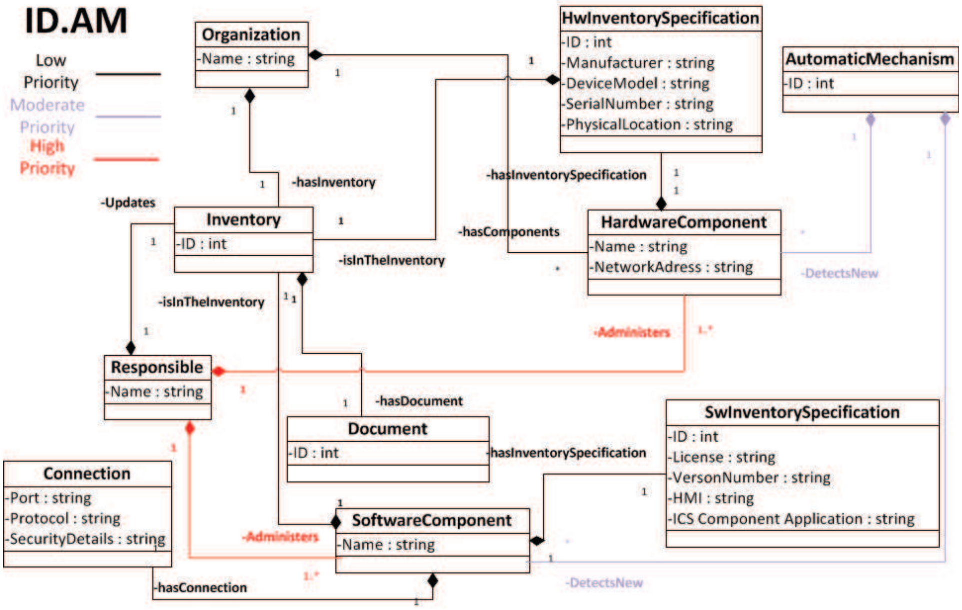
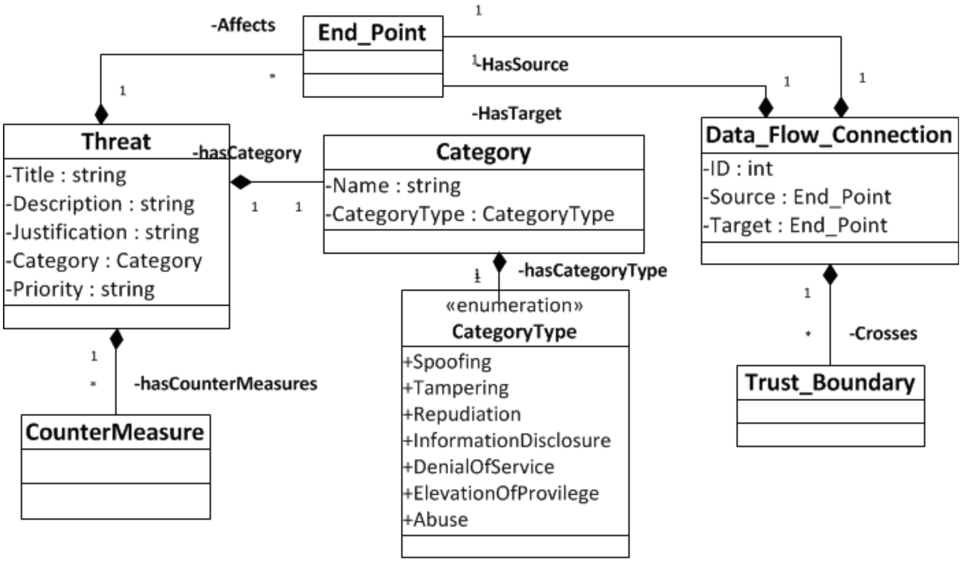


Figure 3.10: Excerpt of the model linking assets with their threats



Chapter 4

4. Virtual Integrative Manufacturing Framework for resources Interaction (VICKI)

This chapter explores a cooperative framework for the interaction of distributed heterogeneous resources in the field of manufacturing, according to the principles of Industry 4.0 paradigm.

4.1 Criticality analysis and requirements elicitation

To define key aspects of the framework and empower its conception, a set of its main requirements are reported in the following.

R1: Context information update. Each resource, after having gathered the information which oversees (e.g. a change of temperature), can affect the knowledge base (context information) shared with the other surrounding resources.

R2: Prompt notification of a context information change to other interested resources. Middleware must propagate contextual state messages to keep the other interested components updated about interesting events (e.g. a task of a working machine has been completed). Notification are distributed as messages, so that the information consumers can perform further evaluations and actions, if needed. Middleware is responsible to both filter the content of each notification message and discriminate the notification target on a per-interest-subscription model. Messages content should be provided to be both automatically interpretable and machine readable. Events notification, as well as their dispatching to other components should adopt a PUSH strategy (providers announce the availability of new information), thus realizing an event-based processing model based on publish-subscribe mechanisms.

R3: Flexible consumers registration to relevant context information. Information consumers should be able to register to specific events simply specifying the kind of information they are interested in (e.g. temperature of a working machine), without bounding to specific knowledge about the producers capabilities.

R4: Supporting private interactions. The middleware must implement a private peer-to-peer messaging policy. Thus, each message can be received by one single consumer, exactly that one subscribed to its content. No information will be lost. Each information authored by a sensor will be retained and made at any time available to any consumer that previously have notified its interest to process that information.

R5: Scalability. The architecture can be easily adapted to the business needs, implying its up- or downscaling according to the changes affecting the subscriptions and also information publication. In particular, this means that computational power and memory can scale on the basis of the production resources and their work load.

R6: Robustness. In a distributed environment, different kinds of runtime problems may occur. For example, components may fail or a network error can occur. Modern industrial application systems have to deal with such disturbances in a self-sustainable way, finding proper solutions for such problems automatically.

R7: Real-time capability. Manufacturing systems often need to provide near-real time services, which are capable to process in an efficient manner structured and unstructured data produced from production resources. Such a capability is more difficult to realize in distributed environments like the IIoT world, due to the complexity overhead emerging from the interaction among distributed components.

R8: Cyber-security. It is essential to identify new security measures that mitigate the risks of internal and external attacks against the manufacturing systems to an acceptable level, thus protecting the information processed by those systems.

R9: Resources discovery and management. In a distributed system it is useful to have a service of yellow page where each device can register their capabilities and functionalities in order to be found by other devices. In particular, the service of yellow pages can allow to look for a service within the IoT ecosystem and then see the details of available registrations. In addition, with the aim to manage the resources, each device can register or deregister some specific services or also only change their descriptions.

4.2 An overview of the framework

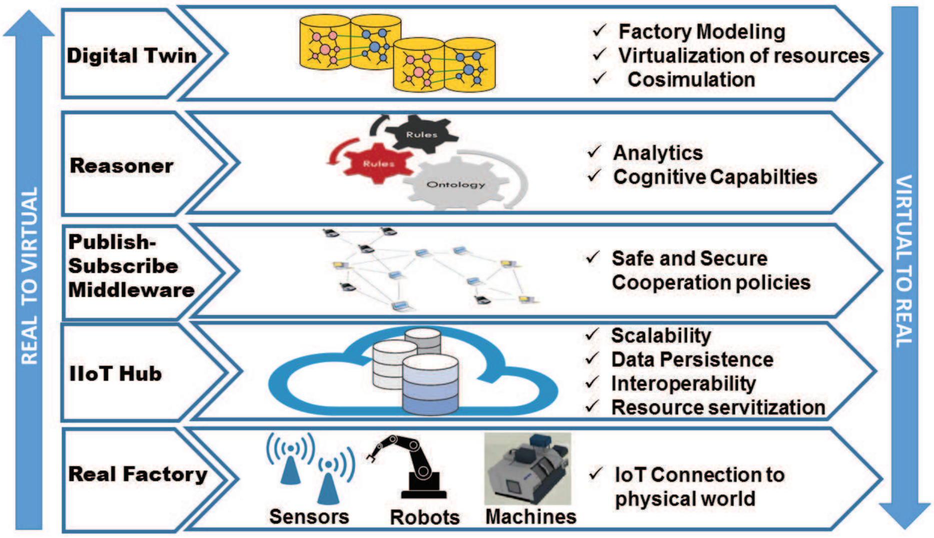
VICKI is a multi-agent and semantic based framework providing near real-time signaling capabilities to all the networked enabled resources involved in a manufacturing system. The applicability of the proposed framework is based on the assumption that a semantic virtual model has been selected to explicitly define the semantics of the factory's objects by taking as a reference a model that all the resources share. Indeed, the development of the semantic model is not the main issue of this study and the model is only a means to demonstrate the feasibility of the overall approach. The reference could be an existing semantic model or a new one designed from scratch.

The design of the framework at the high level leverages the principles of the 5C Architecture for CPS (Lee, Bagheri, and Kao 2015) summarized in Section 2.2. Thus, the framework is designed following a hierarchical architecture structured on the following five main Layers (fig. 4.1): 1) **Real Factory**; 2) **IoT Hub**; 3) **Publish-Subscriber Middleware**; 4) **Reasoner**; 5) **Digital Twin**.

Leveraging these five layers, VICKI allows to distribute new emerging information to interested resources deployed in the real physical environment (**Layer 1, Real Factory**), which can register themselves to be notified about the changing of the state of one or more interesting elements of shared knowledge. Heterogeneous data coming from the shop floor are sent over a cloud-based platform (**Layer 2, IIoT Hub**) where these data are processed to gain useful insights. The cloud platform guarantees the horizontal scalability of the overall architecture; it also exposes the various provided functionalities as micro-services, thus reducing the stratification of the classical conception of the automation pyramid. The evaluation of the emergence of new interesting information is proactively performed by a centralized agency of a middleware (**Layer 3, Publish-Subscribe Middleware**), capable of recognizing an update of the knowledge-base and of notifying eventual updates to subscribers interested in information. The value and the capabilities of physical resources are augmented through their virtualization, achieved under the form of a digital model (**Layer 5, Digital Twin**) that is fully synchronized with the **Real Factory** and thus it represents its true reflection. Leveraging the SWT, **Digital Twin** explicitly defines the semantics of the factory's objects by taking as a reference a model that all the resources share. The reference can be an existing semantic model or a new one designed from scratch.

Moreover, the approach behind the VICKI provides that a set of autonomous and distributed agents act on behalf of the manufacturing system components to allow them to synergistically cooperating to fulfill specific goals. These agents are defined and represented within the model of the **Digital Twin**. In addition, the semantic knowledge base is handled and persisted by a proper RDF Store (Modoni, Sacco, and Terkaj 2014b), a specific database which links and exposes the information through an ad-hoc service (SPARQL endpoint) and hosted on the IIoT Hub. These information are near real-time elaborated by the **Stream Reasoner (Layer 4)**, which can take some preventive or corrective decisions to be applied to the Real

Figure 4.1: The five-layers of VICKI



Factory layer, thus closing the loop between the **Real** and **Virtual world**.

The following sections explores in details the layers of the VICKI.

4.2.1 Layer 1: Real Factory

This layer, corresponding to the Communication level of the 5C Architecture, is responsible for the collection of significant data from machines and production resources, connected through the IIoT protocol. In this regard, different kind of sensors are distributed in the shop floor to measure properties from the real world. The acquired data are then sent under the form of telemetry data flow from the real to the virtual factory, typically leveraging an embedded controller unit enabling sensor data collection and logging (Section 3.1). At this level data are semantically enriched in order to enclose their meaning. In case of legacy devices, they can be retrofitted in order to support the semantics or can ignore the semantics it if the device is a black-box that cannot be upgraded.

4.3 Layer 2: IIoT Hub

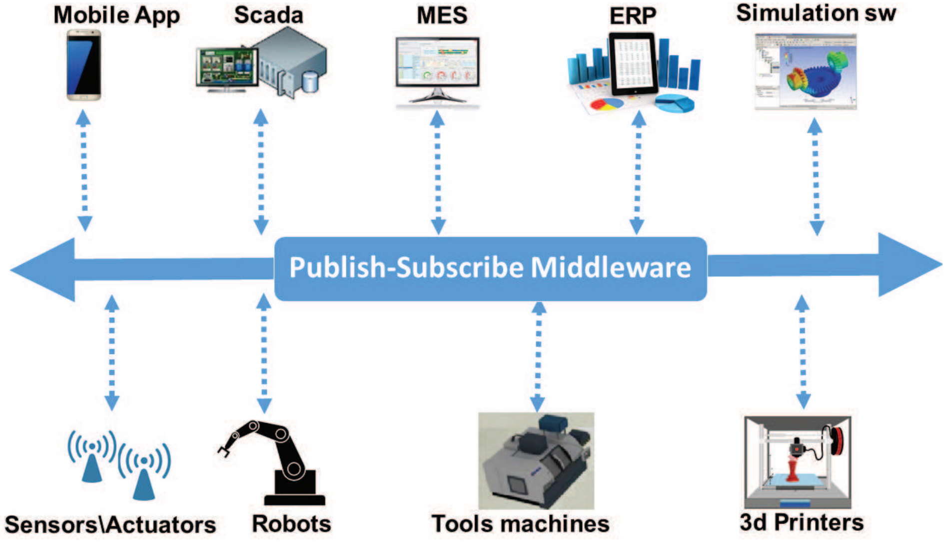
This layer represents the Cloud Gateway and plays the role of central hub where information gathered from every machine connected to the network are collected, persisted and processed. The importance of cloud computing lies in its capability to expose services that can be accessed globally via the Internet. Indeed, all the devices including the legacy systems can provide/consume data to/from the shop-floor via the interface provided by the Cloud layer. This interface supports various protocols (like M2M, ZigBee etc.) and allows to integrate different types of devices. Under these conditions, the **IIoT Hub layer** allows a flexible reconfiguration of the manufacturing resources and the provision of numerous value-added services, thus increasing production efficiency (Mourtzis and Vlachou 2016). In addition, cloud technologies offer scalable and decentralized information processing and data management capabilities. Any database hosted on the cloud to persist the data coming from the shop floor can fully exploit this capability. In fact, the cloud allows to automatically virtualize the available hardware and software resources (e.g. nodes of the server) thus scaling up and down depending on the load of the system. Thanks to its capabilities to enhance interoperability, to ease infrastructure maintenance and to reduce infrastructure costs, cloud is now a widely proven technology within the manufacturing companies. In this regard, companies can choose among a plethora of different commercial solutions that are currently available on the market (e.g. Microsoft Azure (*MicrosoftIoT. Online*), Amazon AWS (*Amazon. Online*), etc.). Among the others, the Infrastructure as a Service (IaaS) paradigm, characterized in that the providers offer computers (physical or virtual machines) and also other resources, can be considered for the IIoT Hub implementation. In order to avoid a data deluge from the shop floor toward the cloud, a crucial phase for the exploration of a valid cloud-based architecture is the study of strategies for distributing intelligence and data between the various components (sensors, microcontrollers, machines, cloud, etc.). To cope with this need, the Edge Computing paradigm can be taken as reference (Shi et al. 2016). This model provides the processing close the data sources and for this reason it is needed a transfer of part of the cloud computing power and storage space near the place where data are generated. This model can contribute to reduce the use of network resources and also the volume of data that have to be elaborated by cloud analytics, thus reducing the transmission time of the data and increasing their availability.

4.3.1 Layer 3: Publish-Subscribe Middleware

The Publish-Subscribe Middleware (PSM) acts as a connector mediating data exchanges between shop floor event sources (publishers) and consumer services (subscribers). Indeed, it provides a central point dispatching information through mechanisms that are transparent to its clients, leveraging an efficient event-driven model based on SWT (Layer 5). In particular, these mechanisms enable a hardware independent interoperability of the consumers, thus allowing to support an easy integration of devices even if based on heterogeneous technologies. As highlighted in fig. 4.2, any resource can act at the same time both as consumer of some type of relevant information and source of other types of information. This proposed solution is more efficient in terms of performance compared to a polling-based model. In fact, being consumers not constrained to poll data through continuous queries to be informed about data updates, the PSM can reduce the bandwidth cost of the semantic queries, as well as the required workload both at the client and server sides.

A relevant feature of the PSM is its capability to express and share a set of

Figure 4.2: Interactions among different clients of the PSM



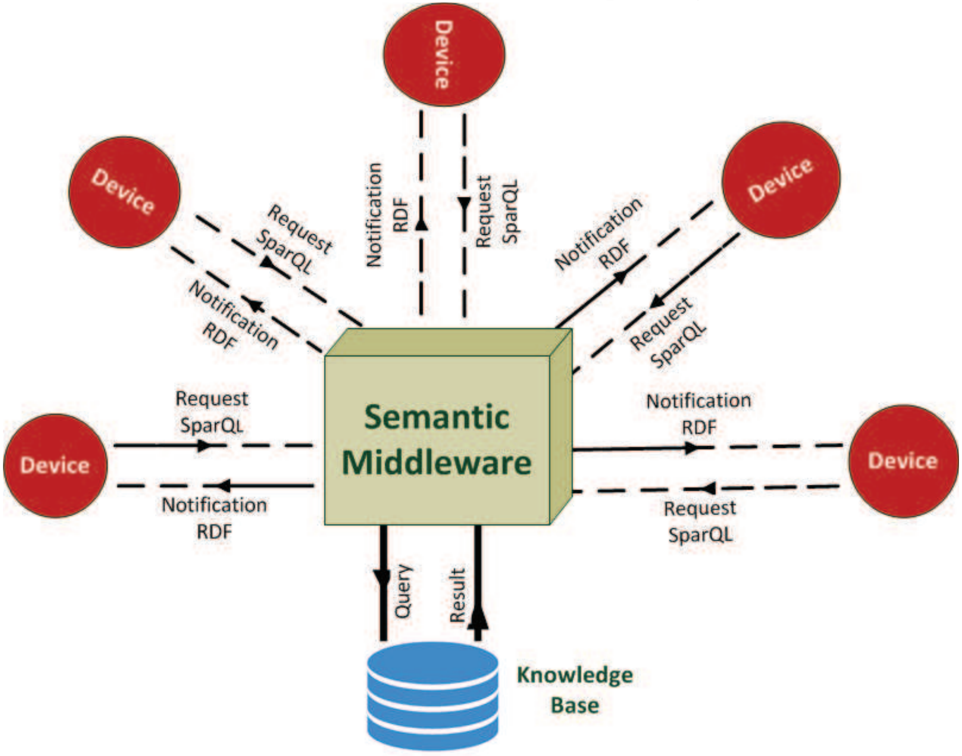
knowledge among the involved resources according to common rules specified under the form of semantic model (Layer 5). In particular, the requests are expressed in SPARQL (*SPARQL. Online*) and for this reason the agents are required to know the SPARQL syntax and the T-BOX structure of the semantic model. In addition, the changed information are notified through RDF (*RDF. Online*). In this way, the connected devices can cooperate synergistically on the basis of a semantic model through which the involved devices can share information. Fig. 4.3 reports the information flows concerning this dispatching service. In the case of legacy resources, where it is not trivial the direct use of the SWT, an automatic semantic enrichment of the data produced by these resources can be carried out through a conversion process from the legacy model to the semantic model (Modoni et al. 2017a).

4.4 Layer 4: Stream Reasoner

This layer processes the information from the shop floor to entail and infer new knowledge about the main concepts of the Digital Twin (Layer 5), in addition to those initially already asserted. Indeed, even the real data stream coming from the real world can be ingested and persisted in ad-hoc databases, typically not all these data are immediately ready to support the decision making process.

The inferred knowledge can be then propagated through the PSM towards the interested resources which can use them to take some decisions to apply appropriate actions. The eventual corrective or preventive decisions that are taken has an impact to the **Real Factory layer**, thus closing the loop between the Real and

Figure 4.3: Semantic Middleware dispatching service



Virtual world (fig. 4.4). In should be also point out that this layer allows logical reasoning in near real time on huge and noisy data streams in order to extract insights (e.g. on the status of the machines). This capability, called stream reasoning, allows to better support the decision process of extremely large numbers of concurrent users (Stuckenschmidt et al. 2010). The reasoning engine is paired with a set of inference rules, which are exploited for the automatic entailments. Rules are formalized taking into account the meta-model representing the Digital Twin of the factory (Layer 5) (fig. 4.5).

The reasoning capability is well suited to specify the dynamics within manufacturing companies. Indeed, typical manufacturing systems functions can be specified and represented in rules. For example, among other applications, it can support the development of complex manufacturing products which involve multiple component dependencies and assembly constraints, that could be specified under the form of part of relationships, rules and other development oriented axioms (Modoni et al. 2017a).

Figure 4.4: The closed Loop between Real and Virtual world

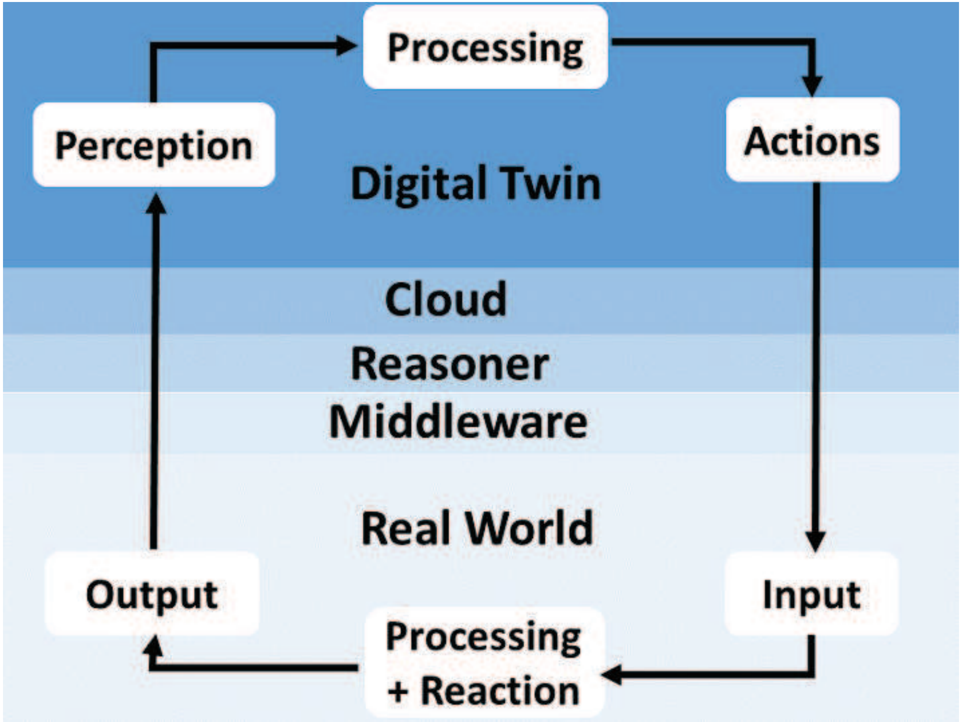
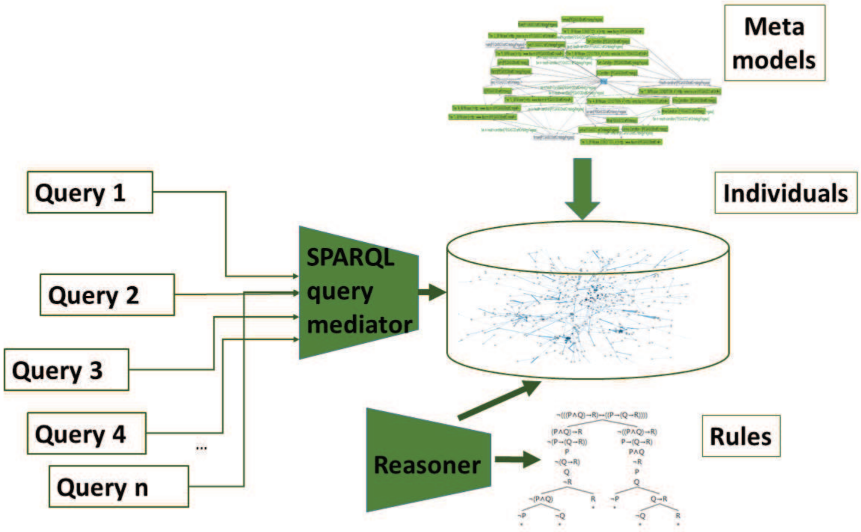


Figure 4.5: Semantic model querying and reasoning mechanism

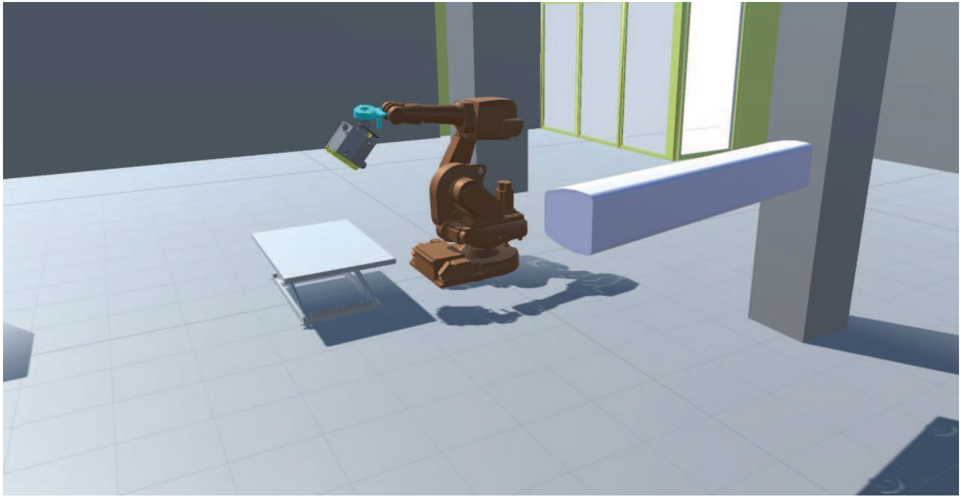


4.5 Layer 5: Semantic Digital Twin

Digital Twin (DT) is a digital image of the real world of the factory, representing various its aspects, such as conceptual, structural and behavioural aspects, according to various levels of detail. DT can be bidirectionally synchronized with the physical world and in this case it represents its true reflection that can be used to simulate system performance, adjust or optimize processes, anticipate failures. In particular, a detailed and updated information about the machining and the degradation of their components can contribute to reduce or even cancel the unplanned downtime, thus increasing the efficiency of the machining process. Moreover, the composition and overlap of a physical resources with its related counterpart containing computed and historical resource information contribute to create augmented entities, which can be used for perform co-simulation within the assembly line. Another advantage derived from the realization of the DT is that it allows the virtualization of the real resources under the form of agents which can act on behalf of the corresponding real counterparts (fig. 4.6 reports the representation of an industrial robot in virtual world). The virtualization can also contribute to enable the dynamic and automatic discovery of resource and of their linked data also across different IIoT ecosystems. The VICKI approach provides that behind each resource there is an agent that can act both as a publisher to send changed information and as a subscriber to receive all the updates related to subscribed information. A main challenge to achieve a comprehensive DT of the factory consists in integrating the heterogeneous data models of the various software and hardware components supporting the whole factory's lifecycle since the early conceptual design phase. At the heart of this process is the modeling of heterogeneous information according to a common shared model. In order to face this challenge, the VICKI approach leverages a rich semantic reference model, that allows to orchestrate the decentralized information, guaranteeing the interoperability between different systems distributed inside and also outside of the plant. In addition, the motivating scenario reported in Chapter 1 justifies the reason to use a semantic model to represent the common knowledge, because the resources that undertake a conversation exchanging messages have to share a common structure of the knowledge exchanged among them.

A specific component provides the function of data mining with the goal to continuously process data streams, in order to extract and derive as much knowledge as possible through specific tools, that combine and transform large amount of structured and unstructured legacy data into semantic data. An information processing algorithm have to be implemented in order to extract large collections of candidate interrelated facts, that include a set of entities, their attributes, and their relations. Unfortunately, converting these candidate facts into useful knowledge represents an arduous challenge that requires to delve into research topics focusing on the problem of knowledge extraction from different sources, including named entity recognition, relation extraction and semantic role labeling (Manco et al. 2017) (Costa et al. 2013).

Figure 4.6: Robot cell in virtual world



It should be noted that DT can be also exploited to strengthen the cyber-security of a manufacturing sytem. Indeed, according to the methodology reported in Chapter 3.3, a proper alignment of the DT with NIST CSF, can help to verify the adherence of the manufacturing system to the NIST CSF principles.

4.6 Discussion

4.6.1 Adherence to Industry 4.0 principles

Table 4.1 highlights how the five layers of the proposed framework are linkrd to the Industry 4.0 principles and technologies.

It should be also enphasized as the framework covers the overall model of the RAMI 4.0 along the three axis, since VICKI connects all the components across the whole factory lifecycle and for all the layers. In addition, it provides a unified model for all the components (from product to Connected World) while there is a consistent data model (the semantic model) during the whole life cycle. In addition, it supports the communication across all IT levels.

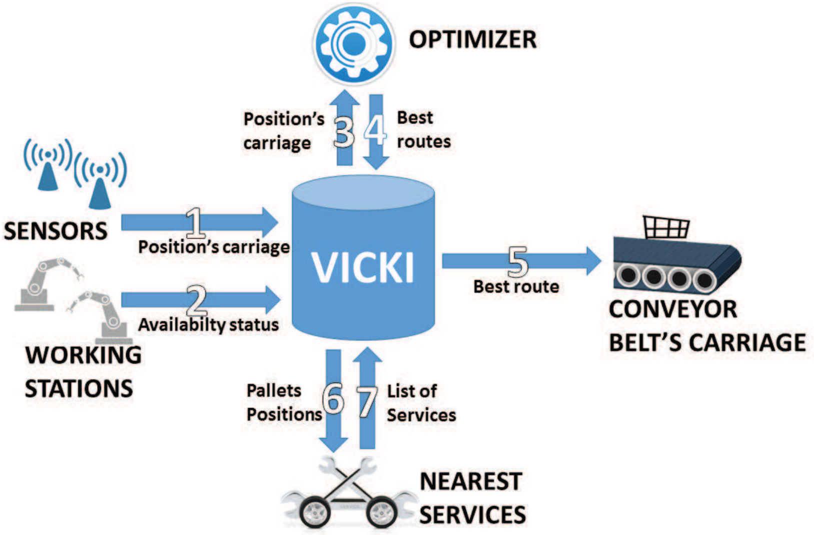
4.6.2 Application to the motivating scenario

Fig. 4.7 points out the VICKI application to the scenario reported in Section 1.4. The sensors monitoring the pallet position will play the role of publisher as they will send the information concerning the pallet position through the middleware (Step 1); this information is expressed under the form of a SPARQL UPDATE. Also the working stations will publish their availability status (Step 2). This information will be then consumed by the simulation tool (Optimizer) which has previously

Table 4.1: Links between VICKI layers and Insustry 4.0 principles and technologies

	Interoperability	**Virtualization**	**Decentralization**	**Real-Time Capability**	**Modularity**	**Service Orientation**
Layer 1: Real Factory			Ubiquitous Computing	Big Data, Factory-Telemetry		IoT
Layer 2: IIoT Hub			Cloud	NoSQL database		
Layer 3: Publish-Subscribe Middleware	Messages Middleware				Multi Agent Systems	Web Services
Layer 4: Stream Reasoner	SWT				SWT	
Layer 5: Semantic Digital Twin	SWT	Multi Agent Systems		SWT	Multi Agent Systems	

Figure 4.7: VICKI application within the motivating scenario



subscribed to the changes applied to the pallet position (Step 3) and the availability status of the working stations (using a proper SPARQL query) with the goal to identify the optimized pallet route. In addition, the information concerning the route is then published (Step 4) and in its turn consumed by the IoT actuators which allow to change the route of the pallets along the conveyor belt (Step 5).

4.6.3 Benefits

It is expected that the adoption of the proposed framework can bring the following strategic advantages:

- **Simplification** of the design and deployment of the overall architecture, since VICKI is built on the basis of several building blocks (e.g. the agents).
- **Complete decoupling** of the event producers (sources) from events consumers. Requests of the data consumers and updates of the data producers are expressed under the form of high-level semantic data.
- **More efficiency** in terms of network use compared to a polling-based solution, since data are sent only when change.
- **Scalability and flexibility** in the production data processing.
- **Flexible addition** of new resources. The reconfiguration of the system can be achieved at run-time, since agents can be added, removed, or also modified on-the-fly, without the need to stop or reprogram the other agents.

- **Incremental transfer technology** from legacy components towards VICKI, since these components can be gradually integrated with WIKI.
- **Flexible adaptation** to the business variation thanks to the underlying easily adaptable semantic model.

Chapter 5

5 An implementation of a VICKI instance: the technology adoption

This chapter introduces *Semantic Event Notifier (SEN)*, a lightweight prototype of middleware that has its root in the VICKI architecture. Its basics were presented in (Modoni et al. 2017b).

5.1 SEN overview

SEN is a multi-agent and semantic based server providing near real-time signaling capabilities to all the networked enabled devices involved in a manufacturing system. It allows to distribute new emerging information to interested devices deployed in the real environment, which can register themselves to be notified about the changing of the state of one or more interesting elements of shared knowledge.

The approach behind the SEN assumes interacting devices or services being loosely coupled agents, synergistically cooperating as a multi-agent system to fulfill specific goals. The evaluation of the emergence of interesting information is proactively performed by a central agency, recognizing an update of the knowledge base, and notifying eventual updates in subscribers' interested information. Under these conditions, this architecture adheres to the Enterprise 2.0 Social Software (E2.0) model (McAfee 2009) (McAfee 2006), supporting social and networked applications to concurrently access and modify a shared knowledge domain.

A key feature of SEN is its capability to express both interests (subscriptions) and following alerts about changed information under the form of semantic data. Requests are expressed as SPARQL 1.1 Query Language (Harris, Seaborne, and Prud'hommeaux 2013) queries and include the expected query result according to the SPARQL 1.1 Recommendations produced by the SPARQL Working Group (*SPARQL Working Group. Online*), allowing each consumer to manage returned information in a best-fit method (Modoni, Veniero, and Sacco 2016). Namely, the returned information are notifiable by means of JSON (Seaborne 2011), CSV or TSV (Seaborne 2013), XML (Beckett and Broekstra 2008) or RDF (Prud and

Seaborne 2006). This way, connected devices can cooperate synergistically by means of a semantic model through which the involved devices can share information, while the middleware supplies the central point which dispatches information through mechanisms that are transparent to their clients. Moreover, SEN includes a graphical components to manage telemetry designed according to the requirements of the application for the factory telemetry (section 3.1.2).

5.2 The implementation

The design of SEN leverages the specifications of the Foundation for Intelligent Physical Agents (FIPA) (*FIPA. IEEE Computer Society. Online*), which include a full set of computer software standards for specifying how agents should communicate and interoperate within a system. Namely, it has been considered the FIPA Subscribe-like interaction protocol (IP) specification (*IEEE Foundation for Intelligent Physical Agents (FIPA) Standards Committee. Online*) that defines messages to be exchanged, as well as their sequencing according to a Request-Reply Enterprise Integration Pattern (Hohpe and Woolf 2004). The FIPA-compliant implementation of the SEN relies on the JADE (Bellifemine, Caire, and Greenwood 2007), a middleware for the implementation of distributed and cooperating multi-agent systems.

SEN is also empowered by a messaging system, supporting both the publish/-subscribe pattern and message queue models. The first allows the specific receivers (subscribers) to express interest in one or more information expressed as SPARQL queries and receive only messages that are of interest. The second enables an asynchronous inter-process communications protocol. In the proposed implementation, the selected messaging system is the Apache ActiveMQ (*ActiveMQ. Online*), an industrial state of the art open source messaging platform providing API in several popular industrial languages. Through the messaging system, clients can subscribe specifying their interests profile, the content type of the expected response and the authorization credentials for the repository (namely, username and password). The communication protocol of ActiveMQ has been set to Openwire (*ActiveMQ OpenWire. Online*). Indeed, unlike other more well-known protocol in the field of IoT such as MQTT, Openwire is capable to support private queues which is one of the essential requirement of SEN. It should be noted that last specifications of OPC UA also provides a publish-subscribe middleware through which clients can subscribe to a specific tag and be notified and informed when the tag's value changes (*OPC UA. Online*). However, this solution also does not provide semantics capabilities. In addition, even one of the most well-known OPC UA implementation based on Microsoft technologies has already implemented this new specification based on publish-subscribe middleware, the proposed solution is totally coupled with Microsoft Azure, while other cloud platforms are not currently supported (*OPC UA net. Online*).

To retain information authored by each sensor (both physical and virtual), they are stored into a RDF store keeping track of the evolutions occurring in the shop floor environment. These information are represented under the form of a semantic model synchronized with the real factory. The RDF store also provides reasoning capabilities to automatically deduce implicit knowledge from the explicitly asserted facts (Modoni et al. 2017a), driven by concepts and relationships interpretation entailments. Moreover, the RDF store is capable of handling large amount of semantic data, also in the form of Big Data (Modoni, Sacco, and Terkaj 2014b). The managed data comprise:

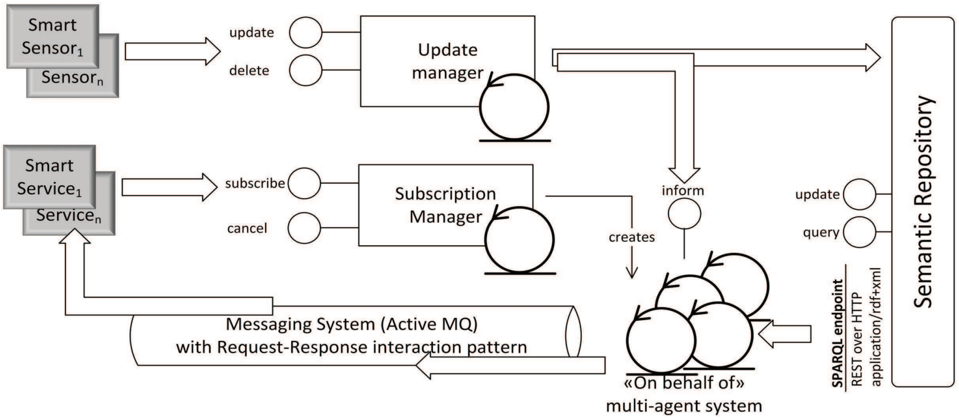
- **the domain ontology**, as representation of the knowledge about shop floor environment;
- **the ontological population**, compliant to the domain ontology;
- **the derivation rules** needed to properly entail the implicit knowledge.

Leveraging the findings of the RDF stores survey reported in Annex 1, as semantic repository it was adopted Stardog (*Stardog. Online*), a commercial solution of Triple Store that well answers the requirements of an intensive data application as the case of the SEN (Modoni, Sacco, and Terkaj 2014b). Moreover, Stardog was deployed to a cloud-based (Armbrust et al. 2010) platform, which guarantees the horizontal scalability of the overall architecture. Finally, it should be noted that the functionality of Resources discovery and management, which is one of the essential requirements, can be implemented using the JADE Direct Facilitator (DF) component, which is also compliant with FIPA specifications.

5.3 The components

Figure 27 depicts the overall structure of SEN, focusing on its semantic information, integration and dispatching capabilities. Referring the more general contexts of Enterprise 2.0 Semantic Knowledge Information systems, SEN is in charge of providing actors interacting with the common knowledge base three fundamental capabilities: a) authoring, allowing information producers to, indeed, author newly explicit information to be shared globally, b) retrieval, as the ability to access and extract information from the knowledge base, and c) signaling, as the capability of being (near) real-time notified about emerging interesting information. This latter can be considered one of SEN's fundamental feature, w.r.t. the requirements of the middleware. The diagram in fig. 5.1 outlines the components in charge of supporting SEN: Update Manager (UM) and Semantic Broker (SB). The latter is on its turn made up by the Subscription Manager (SM) and the Messaging System (MS) supported by a multi-agent system. Each physical or virtual sensor, after having gathered the information which oversees, affects the knowledge base hosted on the shared semantic repository by updating or deleting semantic assertions. This is done through a web service exposed by UM exploiting the Jersey Framework

Figure 5.1: Overall architecture of SEN



(*Project Jersey. Online*) and wrapping up the SPARQL endpoint exposed by the internal Semantic Repository to allow the activation of the MS. On the other hand, information consumers (smart services and sensors) subscribes to the SB, providing their semantically modeled interests' profile.

SM is the component which is always listening on the queue that manages the new subscriptions, leveraging the Apache ActiveMQ (*ActiveMQ. Online*) messaging system. Whenever SM receives a subscription request from a network client, it activates server-side an agent (the ClientAgent) which takes care of the client interests. Namely, SM records this interest activating an "on-behalf-of agent" in charge of signal emerging new information to the consumer. If such agent already exists, SM simply notifies the new consumer's interest. In each moment, the consumer can unsubscribe by cancelling the request. Each time a sensor authors new knowledge, the UM informs the SB of the occurred event so that, in a continuous query processing fashion, the SB can evaluate emerging information and notifies it to the consumer thorough the MS.

5.4 The interaction model

To interact with SEN, standardized subscription and cancellation protocols were defined to which both data providers and consumers must adhere. Fig. 5.2 shows through an UML sequence diagram the interactions among the entities involved in the scenario.

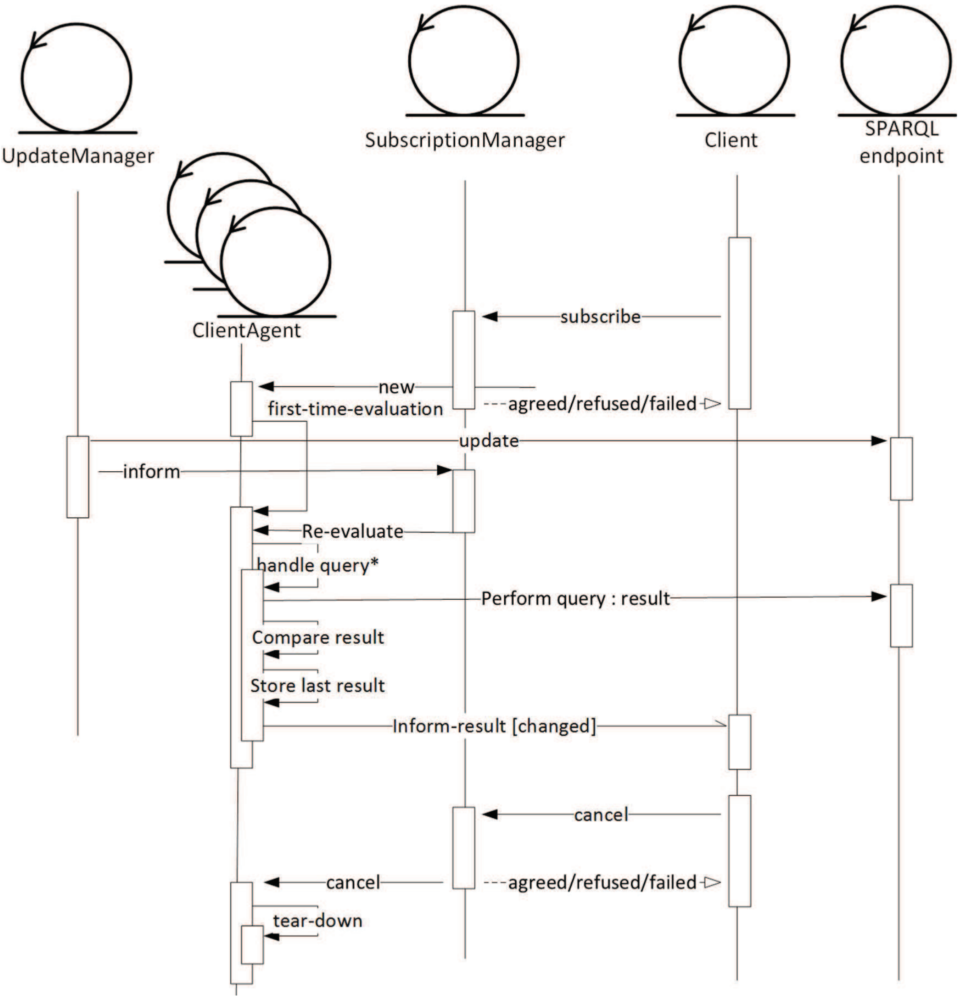
All clients start the interaction with SEN through a subscription message containing the description of the information of interest, the desired response format, an eventually minimum refresh rate in milliseconds (if it desires to be kept up to date even if no information changes), a unique identifier of the request and a reference of the client to which forward the discovered information. The transmitted query should refer the shared knowledge domain described by the semantic model held by the repository. The server processes the subscription request and decides whether to accept it. If it is rejected, the repository sends to the client the rejection condition, ending the interaction. If accepted, the server creates a new on-behalf-of agent or forwards the received subscription to the already living client curator agent. The latter evaluates in its turn the query sending back an information message containing the result of the executed query, compliant with the chosen response format. In case of SELECT, CONSTRUCT or DESCRIBE query forms, the reply is bounded to a not-empty result.

After the transmission of the subscription, the client waits for subsequent notifications generated by its curator agent whenever the knowledge base is changed. When this event happens or on the elapsing of the minimum refresh rate interval, the desired information emerges producing a result that is different from that previously transmitted.

The server continues to broadcast type messages as long as one of the following conditions happen:

1. the client deletes the subscription request by a cancellation request;
2. an error occurs causing the server to be no longer able to communicate with the client or to process queries.

Figure 5.2: SEN interaction model



Chapter 6

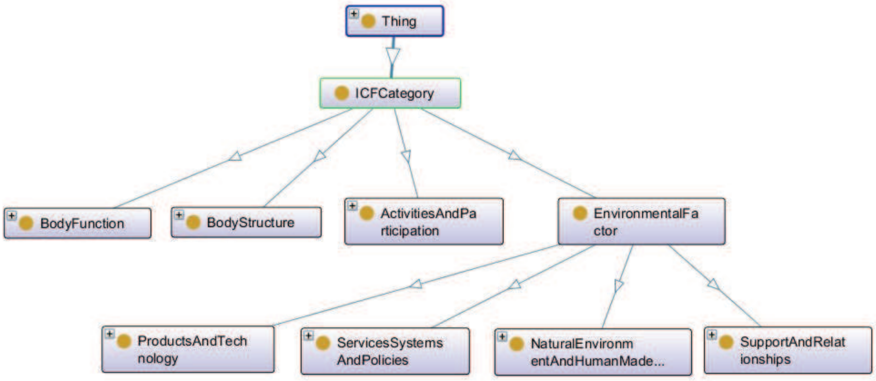
The pilot : an experimental case

The pilot of SEN application has been performed in a field different from manufacturing. In fact, even VICKI has been conceived for a manufacturing system, it is agnostic to the meta model of the used semantic data, and for this reason its transfer technology in other fields is lightweight and can done with little effort. On the other hand we have seen in 3.1 how it is interesting to experiment a transfer of methods and tools from one field to the other where these fields are joined by similar common characteristics. In this section, it is reported a case study where SEN is exploited to support a Cyber Physical System in the field of Ambient Assisted Living (AAL), also leveraging various well-known semantic reference models already available in literature (Modoni et al. 2017b). This pilot has been conducted within the scope of the Design For All (D4A) project (Sacco et al. 2014).

6.1 Design For All project

Design For All (D4A) is a research project co-funded by the Italian Ministry for Education, University and Research within the cluster of initiatives for Technologies for Ambient Assisted Living. It aimed to enhance the semantic interoperability of different devices installed into the domestic environment, so that they can share and exploit the same information. Semantic Web technologies have been adopted to formally describe this information (including the many linking elements) in an ontology, which provides a holistic view of the smart home as a whole, considering the physical dimensions, the users involved, and their evolution over the time (Spoladore, Modoni, and Sacco 2016). It also allows the use of reasoning tools, able to derive new knowledge about the concepts and their relationships, thanks to inferencing rules specified in the Semantic Web Rule Language (SWRL). The semantic model developed, called the Virtual Home Data Model (VHDM), provides a consistent representation of several knowledge domains; it is composed of three main modules: a) the Physiology model, to keep track of users' medical conditions over time; b) the Smart Object Model, which provides a description of

Figure 6.1: A fragment of the taxonomy derived from ICF



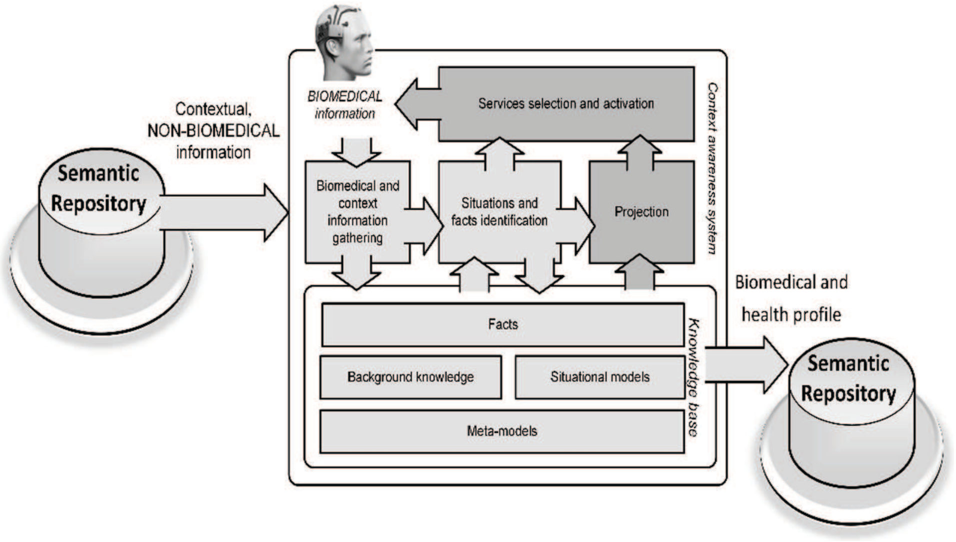
the relationships between appliances and related functionalities; c) the Domestic Environment, which includes information on thermo-hygrometric conditions and air and light quality. A valid starting point for the design of the VHDM has been a study of the various reference models available in the literature covering the relevant knowledge domains. For example, the Physiology model is based on the International Classification of Functioning, Disability and Health (ICF) (fig. 6.1); the Smart Object model refers to the existing Units Ontology, which represents units of measurement and related concepts, in order to support measurements in several fields (e.g. physiological data on the user, environmental data and data related to the smart objects).

6.2 The conducted experiments

One of the defined experimental settings is a context aware Situation Identification System (SIS) designed by abmedica, one of the D4A project partners. The main goal of SIS is the identification of ongoing situations involving one or more observed users, that are relevant from a cognitive point of view in the reference scenario, and the selection of suitable services to be activated basing on the users' contextual and situational profile. The main capabilities of the system are:

- Environment perception and sensing by gathering, analyzing and semantically annotating brain signals, as well as several other bioelectrical and medical information. All of the aforementioned information is merged with contextual information acquired by ubiquitous home-automation sensors;
- Comprehension of the evolving scenario, extracting of all those features useful to identify facts about the observed subjects having a relevance from a cognitive perspective in the reference scenario (i.e. subject characterization as well as relations established with other elements of the scene and currently

Figure 6.2: Situation identification system's architecture



assessed situations). Contextual profiling of the observed user, together with the identification of his status and behavioral patterns are the main results of this activity;

- Projection of the understood scenario with the identification of possibly occurring situations;

- Activation of services to activate according to the current contextual and situational user profile in order to positively affect the environment.

As can be seen in fig. 6.2, the general architecture of SIS recalls the Endsley's model for Situation Awareness (Endsley and Garland 2000) and is based on a specific knowledge base feeding and sustaining situational models both in identification and processing situations. These models and their underlying knowledge are, in their turn, described according to an application ontology focused on modeling and reasoning on medical and behavioral features related to the observed subject. This architectural approach has proven its flexibility and effectiveness in several applicative contexts, ranging from airport security (Furno, Loia, and Veniero 2010) to harbor security (Furno et al. 2011) to ambient intelligence (Mendes, Bizarro, and Marques 2013).

SIS retrieves contextual information coming from smart objects deployed in the environment such as internal and external temperature, light levels, devices activation, users' intentions as feedback of executed tasks, etc. Standard bioelectrical and biomedical information, on the other hand, are gathered by means of standard medical, CE certified, Bluetooth sensors such as blood glucometers, thermometers, sphygmomanometers and heart rate monitors, made smart through a machine-to-machine gateway able to put medical information in the loop. Finally, non-stationary electroencephalographic signals are acquired through an ad-hoc made electroencephalographic wireless helmet, built and owned by abmedica. Gathered signals are analyzed to extract useful features related to the ongoing physiological phenomenon, mental states, closed/open eyes condition, stress level and so on.

During its continuous reasoning process, SIS feeds back to the semantic repository part of the inferred contextual and situational profile, namely the observed users' current state and health profile, thus concurring to enrich the D4A commonly agreed instantaneous profile of the observed user allowing other parties (virtual reality software applications, adaptive user interfaces, and so on) to leverage also on that information to evaluate the premises to execute specific actions to support user (Sacco et al. 2014). Fig. 6.3 depicts the general experimental settings realized in the context of D4A validation.

On the left are represented defined sensors in the role of information producer: a) medical sensors (thermometer, sphygmomanometer, oximeter and heart rate monitor, EEG helmet), together with the information gathering application; b) environment physical sensors (such as light, temperature, humiture, etc.) and virtual sensors, i.e. a comfort evaluation sensor aggregating physical sensors values in a comfort index, once again fed back to the shared knowledge; c) the SIS system, acting both as information consumer and as information publisher(in the role of virtual sensor), providing inferred instantaneous state and health profile of the observed user. On the right, adaptive interfaces and fitness supporting appliances are provided.

All these third-party appliances (sensors, actuators, and services) depend on a prompt discovery of information about the observed user, allowing them to synchronize their internal beliefs and to (near) real-time react to changes occurred in user environment, or user conditions. Given the complex nature of handled information, as well as the potentially huge number of information consumers, a busy-wait PULL-based information discovery model, can rapidly saturate both computational and network resources, understandably soon degrading the overall system performances.

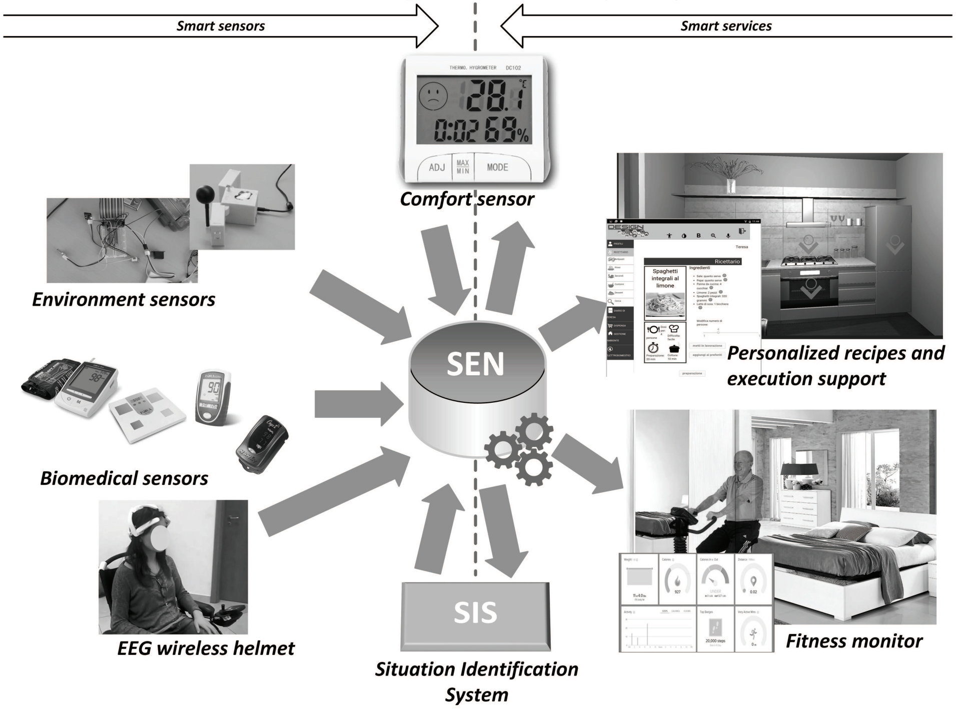
Thus, the efficient PUSH-oriented discovery mechanism provided by SEN was adopted to allow information consumer to promptly discover information relevant for adaptation and control activities.

6.3 The semantic model and the SPARQL queries

According to the aforementioned setting (fig. 6.3) in what follows are listed the information provided by the smart sensors:

- Environmental temperature, humiture and lighting. The sensors, a DS18B20 Temperature Sensor module, a DHT11 Humiture sensor and a Photoresistor module, all of them from Sunfounder and controlled by a Raspberry PC, provide information about temperature, light level and humiture percentage

Figure 6.3: Experimental setting using SIS



of the room where the user is located. The related semantically annotated information is provided with a frequency of 0.017Hz, allowing the system to read environmental information approximately once a minute;

- Environmental comfort evaluation. A virtual sensor gathering environmental sensors' data to provide an aggregated index assessing the living comfort. This sensor acts both as information consumer in that it depends on data instantaneously collected from smart sensors, and as a smart sensor, proving back to the AAL application ecosystem the inferred comfort level;

- Biomedical and bioelectrical data. Measurements are performed before and during the user performs an exercise on a bicycle ergometer, whose activity is monitored by a fitness evaluation system. Namely, measurements relate to temperature, blood pressure (both systolic and diastolic), heart rate. Attention and stress level are evaluated by the EEG helmet. In this case, measurements are performed only before and after the exercise, to avoid artefacts and noise induced by the occurring movement.

Environmental conditions gathered by smart sensors are updated through a "delete and replace" approach, leveraging the adopted Triple Store transactional support to ensure information consistency.

Moreover, a different storing approach was selected for different information types and sources. Contextual environment conditions, such as light levels, temperature, and so on, are stored on an instantaneous-only basis, because in the experimental setting the corresponding historical series was recognized to be useful only to the comfort evaluation sensor. On the other hand, biomedical information is authored and retrieved through a more complex assertions' set, in that subject historical clinical condition was considered somehow relevant for more than a personalization or adaptive system, as well as for design tools leveraging them.

Thus, focusing on light level and temperature and humiture sensors and considering the bedroom as reference environment, information is updated through the following SPARQL Update commands:

```
PREFIX env:  <abox-namespace >
PREFIX envr: <tbox-namespace >
DELETE WHERE {
    env:Bedroom envr:has-property ?t .
};
INSERT DATA {
    env:Bedroom envr:has-property value .
}
```

where *temperature*, *luminance* and *humiture* are the instantaneous floating-point got values, respectively in Celsius degrees, lumen and humiture percentage. This way, the shared knowledge base constantly is aware of the latest instantaneous value for each sensor.

Analogously, an example of how smart services can register for the discovery of these information by means of the following SPARQL Query profile is the following:

```
PREFIX env:  <abox-namespace >
PREFIX envr: <tbox-namespace >
SELECT ?t, ?l, ?h WHERE {
    env:Bedroom envr:has-temperature ?t .
    env:Bedroom envr:has-luminance ?l .
    env:Bedroom envr:has-humiture ?h .
}
```

Biomedical information is authored and retrieved through a more complex assertions' set, in that subject historical clinical condition was considered somehow relevant for more than a personalization or adaptive system, as well as for design tools:

```
PREFIX usr:  <user-abox >
PREFIX msr:  <measurements-abox >
```

```
PREFIX msrr: <measurements-tbox >
PREFIX vsign: <vital-signs-tbox >
DELETE WHERE usr:Donald usr:hasAssociatedMeasurement ?m ;.
DELETE WHERE ?m rdf:type msrTBOX:PhysiologicalMeasurement ;.
DELETE WHERE ?m msrTBOX:hasAssessmentDate ?o ;.
DELETE WHERE ?m msrTBOX:hasDescription ?o ;.
DELETE WHERE ?m msrTBOX:hasMeasurementValue ?o ;.
DELETE WHERE ?m msrTBOX:refers-to vsignABOX:VS-Trouble-condition;.
INSERT DATA .
    usr:Donald usr:hasAssociatedMeasurement msrABOX:Measuree08.
; INSERT DATA .
    msrABOX: Measuree08 rdf:type msrTBOX:PhysiologicalMeasurement .
;.
INSERT DATA .
    MsrABOX: Measuree08 msrTBOX:hasAssessmentDate ”2017-03-
12T22:53:10” .
;.
INSERT DATA .
    msrABOX: Measuree08 msrTBOX:hasMeasurementValue true.
;.
INSERT DATA .
    msrABOX:Measuree08 msrTBOX:refersto vsignABOX:vitalsign.
;.
```

where *vitalsign* can be one of the following biomedical information:

- VS_Systolic_blood_pressure and VS_Diastolic_blood_ pressure, respectively, instantaneous systolic/diastolic blood pressure;
- VS_Glucose_concentration, the measured blood glucose concentration;
- VS_Pulse_rate, the current pulse rate;
- SpO2_concentration, the instantaneous oximetry;
- VS_Axillary_temperature, the axillary temperature of the user.

For each of the authored information, the simplest way for a smart service to discover the instantaneous picture of the user, is by issuing the following SPARQL Query:

```
PREFIX usr: <user-abox >
PREFIX msr: <measurements-abox >
PREFIX msrr: <measurements-tbox >
PREFIX vsign: <vital-signs-tbox >
SELECT ?user ?v WHERE {
```

```
    ?user usr:has-measurement ?m .
    ?m msr:has-value ?v.
    ?m msrr:assessed-on ?date .
    ?m msrr:refers-to vsign:vitalsign
}
ORDER BY DESC(?date) LIMIT 1
```

where *vitalsign* is one of the previously described biomedical information.

6.4 The performance evaluation framework

Performance and scalability are often high priority concerns when selecting a middleware, especially in data intensive application. Moreover, they are among those Information System's features that can be evaluated and assessed through quantifiable criteria, though it is necessary first to understand how to define and implement tests that accurately model the expected workloads while stripping away uncontrollable elements. On the other hand, the proposed middleware has been implemented in the form of a semantic-oriented Continuous Event Processing system. Under this hat, considering the AAL target context, it is recognizable as a good representative of an event-driven application with real-time, mission critical, performance-sensitive constraints, generating huge amounts of data and requiring very short response times to support adaptation and service to user needs. This way a systematic evaluation of its performances, related to its enabling technologies is required. Various benchmarks have been formalized and developed up now in literature; most of them depend on the specific application scenario and are focused on a given functional and performance domain requirements' set. According to (Gyrard et al. 2015), we choose to define a self-made framework that considered the need of automated and timely answers, taking into account the underlying technologies and scenario, while allowing to parametrize performance indicators affecting behavior of both producers and clients.

We agreed to evaluate middleware performances in terms of its ability to notify changes applied to the KB under several conditions. The speed is mainly affected by five major dimensions:

- the size of the knowledge (in triples' count) to work on;
- the complexity of the entailment supporting inference and deduction;
- the size of the derived deduction model (in triples' count);
- the number of working information producers and the update rate;
- the number of subscripted consumers and the complexity of required knowledge graphs.

When dealing with information emergence identification or querying, the first three issues mainly affect the size of the model making up the knowledge base, and consequently the time required to find the triples graph matching the information modeled by the discovery profile. Indeed, the complexity of the entailment introduces a further issue w.r.t. performance evaluation, bound to the time required to be applied to an eventually huge KB before executing any query. This, on its turn, depends on the chosen algorithm and the size of the explicit assertions making up the KB itself. Therefore, we decided to remove this element to obtain a simpler model.

The benchmark consists in processing simultaneously several parameters affecting the system's workload:

- $K \subseteq \mathbb{N} - \{0\}$, the size of the knowledge base to work on, in terms of triples' count;
- $S \subseteq \mathbb{N} - \{0\}$, the number of working sensors;
- $C \subseteq \mathbb{N} - \{0\}$, the number of subscripted consumers (smart services or sensors) under a set of queries;
- $Q \subseteq \mathbb{N} - \{0\}$, the number of subscriptions for each consumer.

Because the size of knowledge base is the main reference parameter, we combined described parameters by means of the following weighted expression:

$$\alpha\xi_{k_M}(k) \cdot (1 + \beta\xi_{s_M}(s)) \cdot (1 + \gamma\xi_{c_M}(c)) \cdot (1 + \delta\xi_{q_M}(q)) \tag{6.1}$$

where

- $\alpha \in]0 \dots 1]$, $\beta, \gamma, \delta \in [0 \dots 1]$ are constants representing the relevance of each parameter in the problem's space. Constraining α to be a positive value we recognize that the size of the knowledge base is the main affecting parameter;
- $k_M \in K - \{1\}$, $s_M \in S - \{1\}$, $c_M \in C - \{1\}$, and $q_M \in Q - \{1\}$ are, respectively, $maxK$, $maxS$, $maxC$ and $maxQ$.

Finally, said P one of the identified performance affecting parameters' values set, for any $\rho_M > 1$ and $\rho \geq 1$, $\xi_{\rho_M} : P \to [0 \dots 1]$ is a normalization function defined as

$$\xi_{\rho_M}(\rho) = \frac{ln(\rho)}{ln(\rho_M)} \tag{6.2}$$

ξ is aimed to balance the effects of different increment schemes foreseen for each basic parameter. Note that $0 \leq \xi_{\rho_M}(\rho) < 1$ for any pair ρ, ρ_M.

The reference load factor is given by the size of the knowledge base, while the other parameters are considered magnifying factors, with a specific magnification impact. In our experiment, we set $\alpha = 1$, $\beta = 0.5$, $\gamma = 1$ and $\delta = 0.7$. With this assumption, we want to stress the fact that the main affecting parameters are the size of the knowledge base and the number of subscripted consumers. Indeed, sensors influence the number of signaling activations depending on uncontrollable scheduling issues. In this sense, more authoring activities can be masked by a single following signaling, being thus collapsed in a single action. At the same time, the number of subscripted queries affects response time because of the serialized management approach adopted to serve a specific subscriber needs.

Considering the described parameters, each of them can assume values in a different range. Furthermore, each of them may have a generally different increasing step, and different impact on the overall performance when ρ approaches ρ_M on higher order values. For instance, triples' number grows very fast, but the difference of the impact on performances of a given triples' number and the immediately next increase may be negligible. At the same time, the difference of the impact on performances between a low triples' number and a high one is generally very significant. On the other hand, this does not generally hold when considering one of the remaining dimensions. These thoughts lead us to consider we need to consider some tailoring when normalizing over the several measured dimensions. The increasing step is introduced in the normalization function, together with a specialized logarithms' root for each of considered parameter, allowing to specialize the growing rate of logarithm while enhancing the differences between low and high values.

def. 1 *Given P a performance parameter, calling* i_P *the increase step for P, and* r_P *the root logarithm used to normalize values over P,* $\xi^P_{\rho_M} : \mathbb{N} - 0 \rightarrow [0 \ldots 1]$ *is a normalization function defined as*

$$\xi^P_{\rho_M}(\rho) = \frac{log_{r_P}(\frac{\rho}{i_P} + 1)}{log_{r_P}(\frac{\rho_M}{i_P} + 1)} \tag{6.3}$$

where ρ_M *is the maximum value of* ρ *in P.*

In the conducted experiment we set

- $k_M = 10^6$, $i_K = 5 \cdot 10^4$ and $r_K = 1.2$ (thus selecting a more rapidly increasing logarithm for triples' number normalization than the other ones, to enhance differences with them);
- $s_M = 10^1$, $i_S = 1$ and $r_S = e$;
- $c_M = 5 \cdot 10^2$, $i_C = 5$ and $r_C = e$;
- $q_M = 5$, $i_q = 1$ and $r_Q = e$.

Finally, it naturally results that

$$\begin{aligned} 0 \leq \alpha\xi^K_{k_M}(k)(1 + \beta\xi^S_{s_M}(s))(1 + \gamma\xi^C_{c_M}(c))(1 + \delta\xi_{q_M}(q)) \leq \\ (1 + \beta)(1 + \gamma)(1 + \delta) \end{aligned} \tag{6.4}$$

Eq. 6.3 concurs to define the load of the system, following the definition below.

def. 1 *given* k_M, s_M, c_M, *and* q_M *respectively representing the upper limit to the size of the knowledge base, the number of working sensors, the number of subscripted consumers, and the number of subscripted queries per consumer, the load of SEN is given as the function* $load : K \times S \times C \times Q \rightarrow [0..1]$ *defined as*

$$load(k, s, c, q) = \frac{\alpha \xi^{K}_{k_M}(k)(1 + \beta \xi^{S}_{s_M}(s))(1 + \gamma \xi^{C}_{c_M}(c))(1 + \delta \xi_{q_M}(q))}{(1 + \beta)(1 + \gamma)(1 + \delta)} \tag{6.5}$$

In def. 1 the logarithmic scale allows to consider a slowly increasing load, especially when dealing with great input sizes, to flatten local differences.

Fig. 6.4 depicts the loads' surface obtained varying performance parameters. As mentioned earlier, the purpose of the benchmark is to evaluate the ability of the middleware to process increasing loads while providing quick answers with low latency. Latency is commonly agreed as the time interval between a stimulation and a response in a system (or, from our perspective, the time delay between a change in the knowledge base and the notification of emerging information to the client).

To evaluate performances, here we interpret latency as the stimulus turnaround time or *stimulus processing turnaround*, i.e. the average time elapsed between a new information gathering and reporting the result back to all involved subscribed consumers, under a well-defined load. To filter noise induced by several uncontrollable factors (network latency, operating system scheduler, and so on), for each instance of a performance evaluation, the trial was repeated several times and the resulting latencies averaged using the harmonic mean ($latency_m$), thus giving a measure that is more robust in the presence of outlier values than other statistical means such as mean or median. W.r.t. the described experimental setting, the trials count was set to 10.

Finally, load under a given configuration *t=<k,s,c,q>* and the average latency are combined to define a synthetic evaluation parameter named *performance score* and denoted by p^t_{SEN}, as follows.

def. 2 *provided a trial configuration* $t =< k, s, c, q >$ *for the set of trials T, and given* $load_t$ *and* $latency_{m,t}$, *respectively representing the load induced by the trial configuration t and the average latency of stimulus processing turnaround, the score reached by SEN is given by* $p^t_{SEN} : \mathbb{R}^+_0 \rightarrow \mathbb{R}^+_0$ *defined as*

$$p^t_{SEN} = load_t \cdot \frac{max_T(latency_{m,t})}{latency_{m,t}} \tag{6.6}$$

where $max_T(latency_{m,t})$ *is the maximum* $latency_{m,t}$
measured on all possible trials.

Note that p^t_{SEN} has essentially a comparison purpose aimed to understand if SEN performances have a constant trend, w.r.t. to load affecting parameters.

To perform the evaluation a test framework has been developed as illustrated in fig. 6.5.

Initially, the user specifies the work parameters k, s, c and q, together with its preferred relevance or, alternatively, uses the standard benchmark configuration to create a test setup (Step 1). Then, the Benchmark Generator module generates a *simulated environment* (Step 2) made up by the desired instances of both producers (smart sensors) and consumers (smart services), the former notifying the desire information to SEN, and the latter subscribing to its *signaling services*. The generated environment (both sensors and services) is then activated (Step 3) to start performances measurements. During the execution lifetime, each Stimulus-Reaction Correlation (SRC) is logged by SEN and transferred (*performances data flow*) to the Performances Collector. SCRs allow to both track input-output relations and tear down latency correlations. Finally, performances are aggregated and reported to the user (Step 4) as tables and diagrams needed to assess performances. All tools provide an easy to run environment and thus they require very little effort to be executed.

6.4.1 Results of the assessment

In this section we report the results about SEN's deployment in the real-world case study introduced in 6.2. The main significant slice of over one thousand performance tests has been reported in figures from fig. 6.6 to fig. 6.14, where both the knowledge base size and the registered consumers change over the respective experimental ranges. Other reports have been omitted here because the other dimensions do not add a relevant value, being analogous to the formers.

Other reports have been omitted here because the other dimensions do not add a relevant value, being analogous to the formers. More precisely, figures from fig. 6.6 to fig. 6.12 present the details of performance tests executed under several conditions and evaluated according to eq. 6.6 compared to the system load defined by eq. 6.5.

Each figure reports on the left the system load (the lower graphic) and the evaluated p^t_{SEN} (the upper graphic). On the right side, instead, is reported the $latency_{m,t}$ in milliseconds, as the whole stimulus-receipt round-trip-time (including the time required by the access to the knowledge base, averaging around 60-100ms). Both sides adopt the size of the knowledge base in triples' count as abscissa that ranges in $5 \cdot 10^4 \leq k \leq 10^6$.

fig. 6.13 and fig. 6.14, instead, show, respectively, an aggregated report allowing to point out how SEN performances scale better upon the growing of system's load. As it can be seen, the system scores better under high loads, generally following its growth rate. Moreover, when the size of the knowledge base goes towards huge loads, the systems responds positively with a constantly growing score jumps significantly upwards. This seems to be true under any stress condition, as shown by fig. 6.13. At the same time, fig. 6.14 points out how the execution time tends to stabilize driven by the number of concurrent consumers expecting a reply.

Figure 6.4: SEN load focusing on triples' count as main parameter (right coordinate, with a 10^3 unit) and, from top to bottom, respectively, subscribed consumers, queries per consumer and sensors as left coordinate

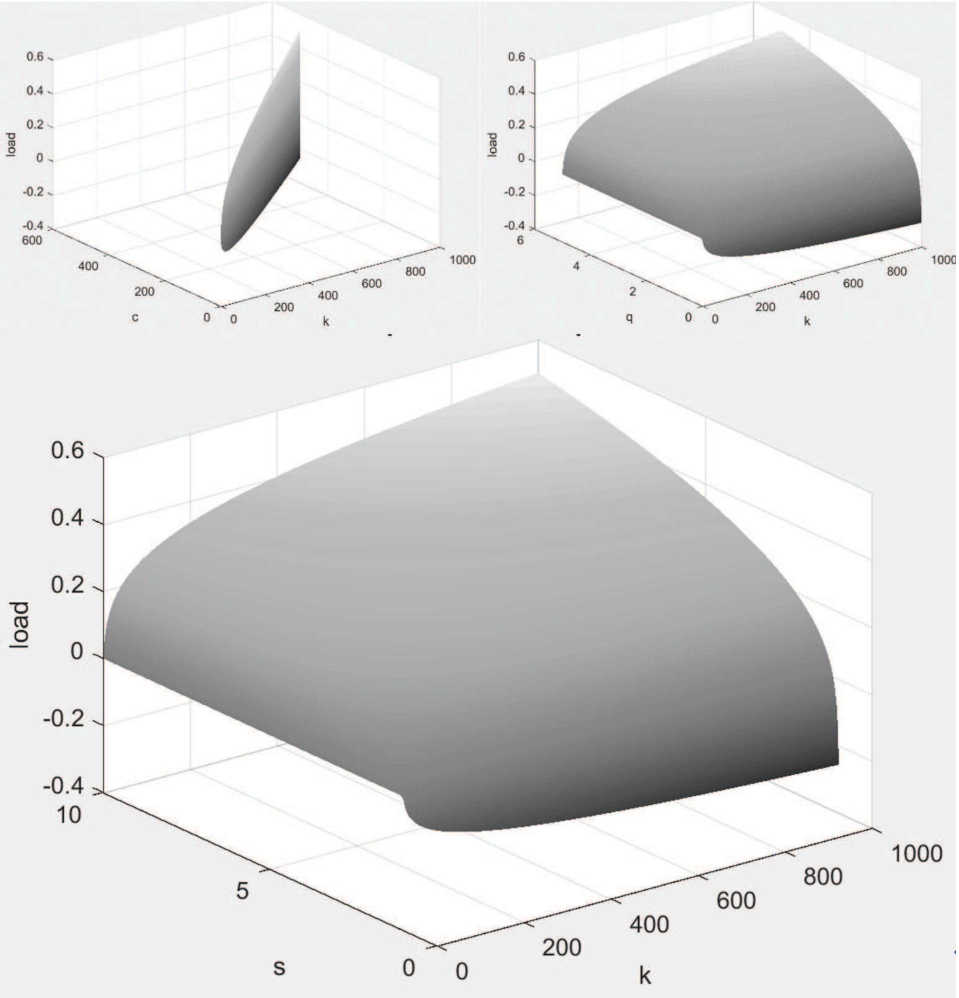


Figure 6.5: Testing framework

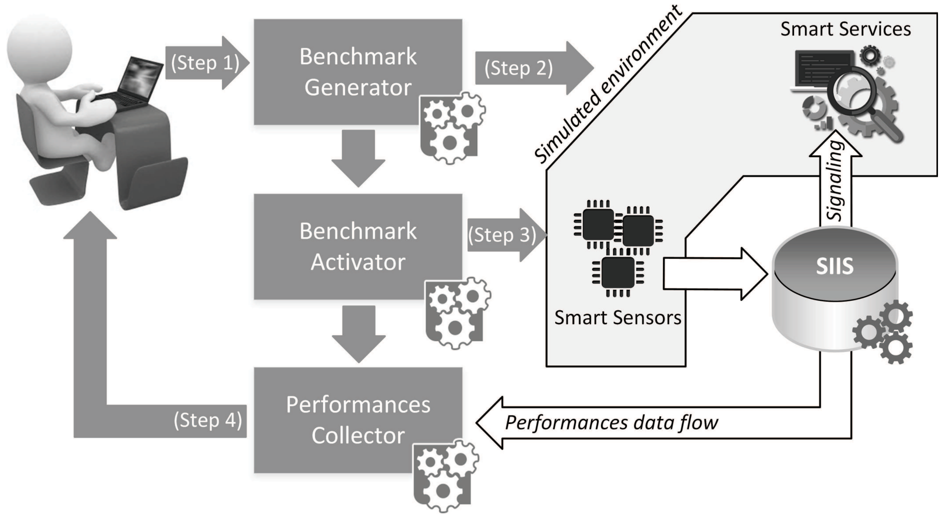


Figure 6.6: p^t_{SEN}, $load_t$ (left) and $latency_{(m,t)}$ (right), $t =< k, 1, 1, 1 >$

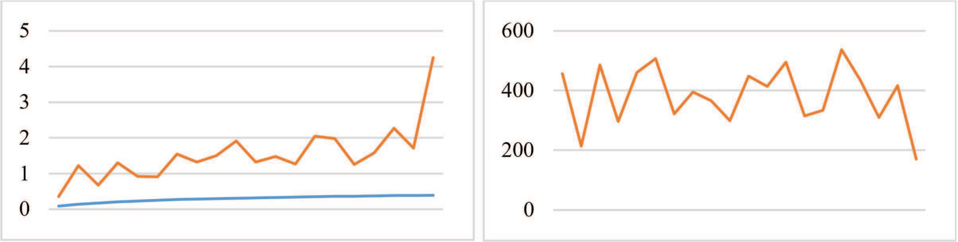


Figure 6.7: p^t_{SEN}, $load_t$ (left) and $latency_{(m,t)}$ (right), $t =< k, 1, 6, 1 >$

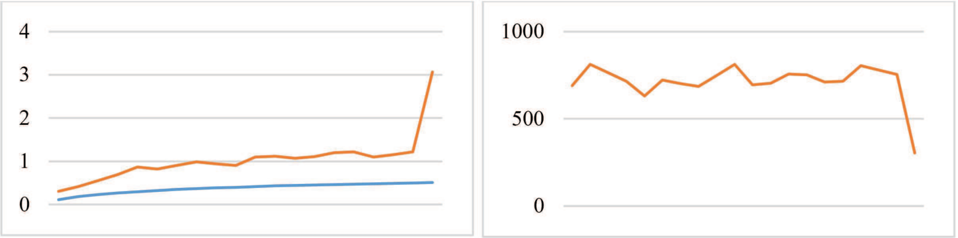


Figure 6.8: p^t_{SEN}, $load_t$ (left) and $latency_{(m,t)}$ (right), $t =< k, 1, 11, 1 >$

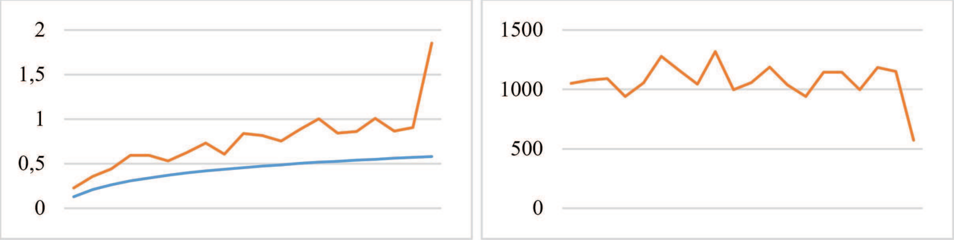


Figure 6.9: p^t_{SEN}, $load_t$ (left) and $latency_{(m,t)}$ (right), $t =< k, 1, 16, 1 >$

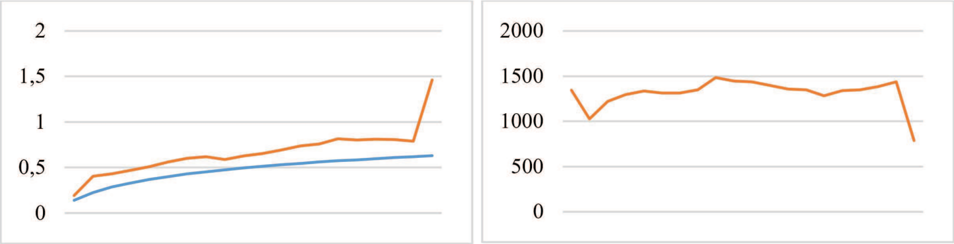


Figure 6.10: p^t_{SEN}, $load_t$ (left) and $latency_{(m,t)}$ (right), $t =< k, 1, 21, 1 >$

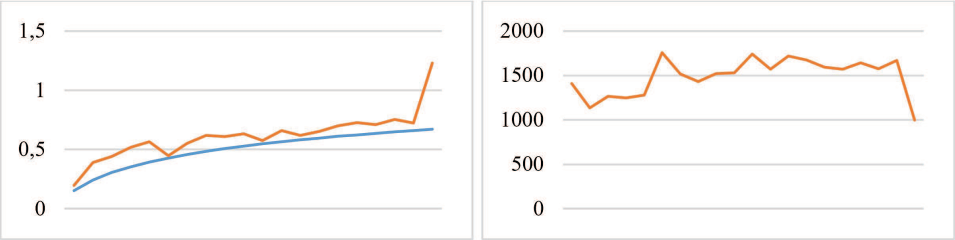


Figure 6.11: p^t_{SEN}, $load_t$ (left) and $latency_{(m,t)}$ (right), $t =< k, 1, 26, 1 >$

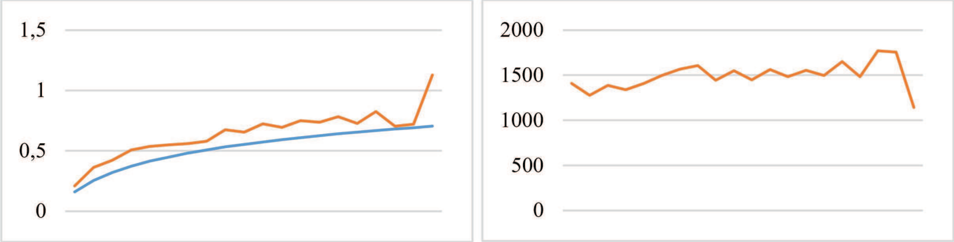


Figure 6.12: p^t_{SEN}, $load_t$ (left) and $latency_{(m,t)}$ (right), $t =< k, 1, 31, 1 >$

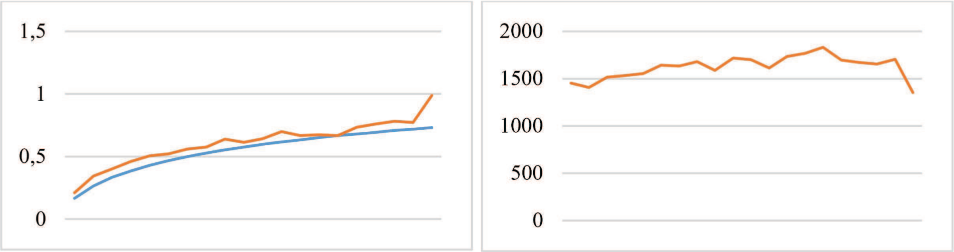


Figure 6.13: p^t_{SEN}, with $t =< k, 1, c, 1 >$ where $5 \cdot 10^4 \leq k \leq 10^6$ and $1 \leq c \leq 31$

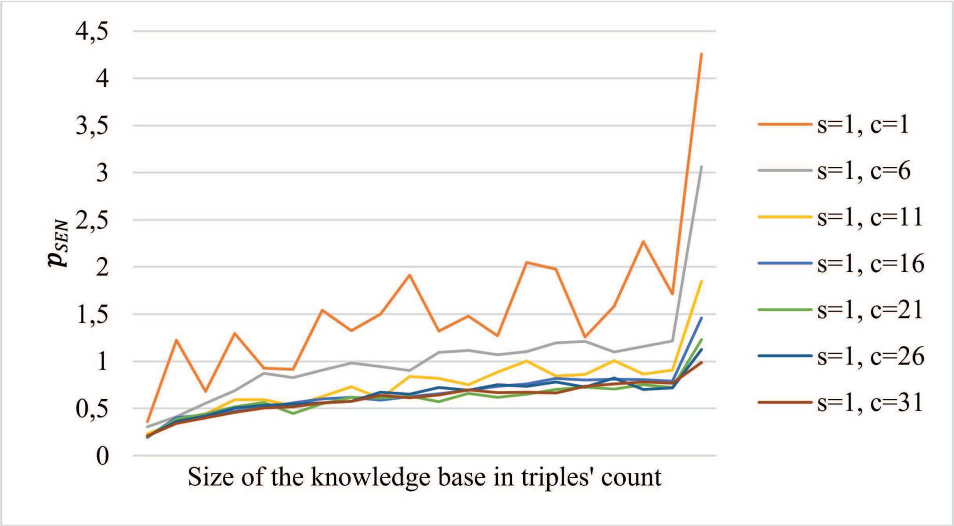
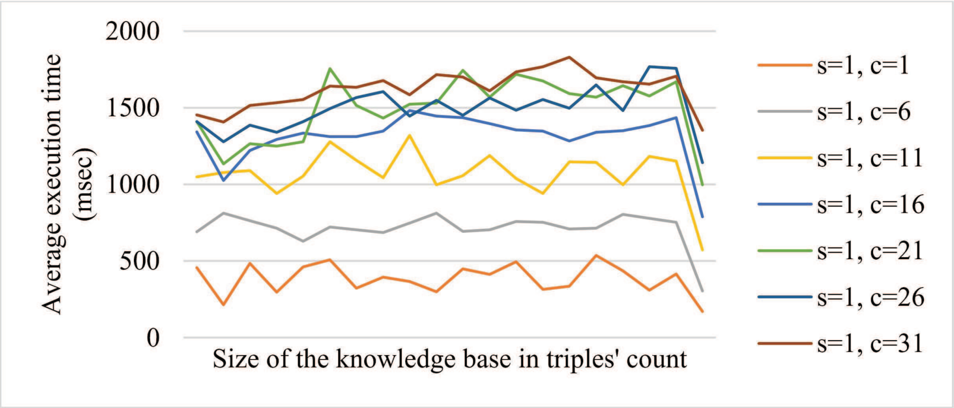


Figure 6.14: Average execution time under several conditions



Conclusions

This doctoral thesis work has explored a cooperative pattern for the interaction of distributed heterogeneous resources in the field of manufacturing, according to the principles of Industry 4.0 paradigm. A major result of this research is the Virtual Integrative manufaCturing frameworK for resource Integration (VICKI), a reference architecture for context-aware intelligent manufacturing application. The framework enables a model of communication and collaboration among manufacturing resources which allows to dynamically propagate the information changes occurred within the shop floor towards all the interested and enabled components. The exploitation of VICKI in manufacturing can enhance and integrate the orchestration of the services supporting the whole lifecycle of the factory, thus giving a significant contribution towards the implementation of a platform of interoperable services as provided by the Industry 4.0 vision. In particular, the framework functionalities can be exploited in all the fields where the cooperation among resources is a strict requirement. One of these fields is the intelligent production, where the framework functionalities can enable smart product order in order to autonomously control its moves along the production line. Another field of application can be the collaborative robotics where VICKI can be an enabling tecnology for the cooperation among robots and humans. In addition, the framework can be useful in other fields different from manufacturing (e.g. to support the interaction of autonomous cars). In fact, even it has been conceived for a manufacturing system, it is agnostic to the meta model of the used semantic data, and for this reason its transfer technology in other fields is lightweight and can done with little effort.

Leveraging an AAL real case study, we showed that heterogeneous devices can be integrated into a IoT based system and configured to share knowledge through a system application that is based on the VICKI framework. The conducted pilot showed the correctness of the proposed approach, allowing to demonstrate a set of benefits, mainly in terms of scalability of the overall architecture. Specifically, the obtained performances of the implemented system are very encouraging. The implemented system is capable to support an high number of connected devices (up to 100) while sharing a large knowledge-base (up 106 semantic triples). All the achieved outcomes represent a valid starting point which still needs further enhancements. In this regard, next section reports major open issues on which it will be crucial investigate.

Future steps

After the PhD, future studies and developments related with this work will mainly address four goals. These future works of consolidation will be mainly carried out within the ongoing developments of the INTER-IoT project, which is strictly linked to the PhD work.

The first goal will regard the exploration and implementation of a Trust model to enable reliable and secure interactions between trustworthy entities. For example, in the motivating scenario reported in 1.4, the actuators must be sure that sensors monitoring position of the pallet are trustworthy before consuming these data. In this regard, a potential solution of Trust model is based on the evaluation of the devices reputation (Bossi, Braghin, and Trombetta 2014), so that only the devices that are granted can publish critical information.

The second goal will regard the optimization of the queries execution, through a smarter organization of the meta-model based on a subdivision in different graphs. This way, when a change is triggered, the system will reload only the queries involving the graphs affected by the changes, and, at the same time, the system could provide a real upper bound to the memory consumption induced by per-client agents activation. Moreover, while the current implementation assumes that there is a semantic meta-model shared between the different devices, the future implementation will investigate the capability to allow each device to have its own semantic model, leveraging modules for ontological alignment of different models (Cudr-Mauroux 2013).

Finally, fourth goal will concern the identification of a valid pattern of edge-level processing to filter the noise, remove the corrupted data and aggregate information, thus contributing to minimize the congestion of the network.

Appendix A

A survey of RDF stores

The purpose of this section is to present a qualitative comparison of a set of RDF store solutions. The study, based on the available literature, aims to identify a valid backbone to support large-scale semantic applications in the context of enterprises facing the challenge of realizing a more integrated management of the innovation process (Modoni, Sacco, and Terkaj 2014b).

A.1 Preselection of RDF stores to be evaluated

As the storing, accessing, and processing of large amount of data represents an essential requirement in order to satisfy the needs to search, analyze and visualize these data as information, the capability to manage a large number of triples (greater than 2 billions) has been chosen as filter criterion for selecting a set of stores to be evaluated. Based on this filter, it has been selected a list among the most well-known RDF stores that satisfy this requirement, excluding all technologies which are discontinued or in a too early state of development. The list is reported below; for each item it is also reported a brief overview.

AllegroGraph[1] is a closed source RDF store developed by Franz Inc. in Common Lisp. Currently it is in use in various open source and commercial projects. It is the storage component for the TwitLogic project that is bringing the Semantic Web to Twitter data.

Blazegraph[2] is a platform developed by SYSTAP LLC, that can be deployed as an embedded database, a standalone server, a replication cluster, and as a horizontally-shared federation of services similar to Google's bigtable, or Cassandra.

---

[1] `http://www.franz.com/products/allegrograph/`

[2] `https://www.blazegraph.com/`

Apache Marmotta[3] is a platform implemented as a Service-Oriented Architecture (SOA). Currently, the following backends are available: KiWi Triple Store, Sesame Native and BigData (experimental).

OpenLink Virtuoso[4] is developed by OpenLink Software as a database engine hybrid that combines the functionality of a traditional RDBMS, virtual database, RDF, XML in a single system. It is used to manage data exposed by dbpedia.org.

Oracle 12c[5] supports RDF data management with his component Oracle Spatial and Graph RDF Semantic Graph. It is a commercial product, which is free of charge for non-commercial purposes. There are some examples of use in integrated bioinformatics, health care informatics, finance, web social network, and content management.

Stardog[6] is a graph database based on pure Java storage and designed for mission-critical applications.

A.2 Comparison of RDF stores

This subsection presents the results of the comparison based on four criteria and carried out on the set of preselected RDF stores. To facilitate the analysis of the technical characteristics, a synopsis of main outcomes of this study is reported in table A.1. In the following it is presented, for each criterion included in the adopted framework, the evaluation of the selected RDF stores.

Handling streaming data. Very little information is available at this time about how the surveyed RDF stores act when they handle streaming data. AllegroGraph has been involved in an experimental evaluation of an architecture serving as a real-world example of a high-performance RDF streaming application that sends high-throughput updates to remote store in an Internet-scale distributed environment. There are examples of Stardog's use as backbone of software application for near real-time representation of situational knowledge acquired, also at high frequencies, from heterogeneous sensor network data.

To understand how an RDF store is able to cope with huge flows of near real-time information coming from many sources, it is necessary to analyze his performance profile in such a context and verify if it satisfies the expectations.

Several metrics can be used to facilitate a standard and systematic evaluation of the RDF stores performance when dealing with huge flows of near real-time

---

[3] `http://marmotta.apache.org/`

[4] `http://virtuoso.openlinksw.com`

[5] `http://www.oracle.com/technetwork/database/enterprise-edition/overview/index.html`

[6] `http://stardog.com/`

Table A.1: Evaluation synopsis of a set of technical characteristics

AllegroGraph	Commercial, Free	Windows, Linux, Mac	Lisp	Native (Graph)	Java, Python, Lisp, Ruby, Perl, C#	SPARQL
Blazegraph	Commercial, GNU GPL	Linux / Unix	Java	Hybrid	Java	SPARQL
Marmotta	Apache2	Windows, Linux, Mac	Java	Hybrid	Java	SPARQL
Openlink Virtuoso	Commercial, GNU GPL	Windows. Linux, Mac	C	Hybrid (object/relational)	Drivers for ODBC, JDBC, ADO.NET and OLE DB	SPARQL, SQL
Oracle 12c	Commercial, Free	Linux, Windows, Solaris	Java	DBMS backed	Connectors for SQL and Java-based applications	SPARQL, SQL
Stardog	Commercial, Free	Windows. Linux, Mac	Java	Native (Graph)	Java, Ruby, Python, .Net	SPARQL

information coming from many sources; two of the most commonly used metrics are query duration and load time. Query duration is the amount of time taken to return the result set for a specific query. Load time is the amount of time taken to add some information to the store, including any overhead occurring as part of this operation. Other useful metrics include disk space requirements, memory footprint and deletion duration.

Based on the above metrics, several benchmarks (e.g. (Guo, Pan, and Heflin 2005), (Bizer and Schultz 2009), etc.) have been formalized and published. However, they do not always allow a rigorous evaluation and, moreover, do not fully satisfy all the successful requirements of a good benchmark (i.e. relevant, repeatable, fair, verifiable and economical) (Boncz et al. 2013). Therefore, a definitive evaluation of RDF store performance is still missing. Another interesting study is within the EU project LDBC (Linked Data Benchmark Council) (Boncz et al. 2013) that joined together a community of academic researchers and industry, whose main objective was the development of open source and industrial grade benchmarks for graph and RDF databases, leading to an easier comparison of the different technologies also in the context of scenarios managing massive amounts of streaming data.

Security. Virtuoso is based on a graph-level role-based security policy; thus, users can be granted permissions to specific graphs, either read-write or read-only. Stardog's security model is also based on role-based access control at graph-level: in this case users can be granted permissions over resources to which access is to be controlled. Moreover, in order to provide better data protection, it supports extensible authentication via both internally stored user information and external mechanisms (e.g. LDAP) and in-flight encryption of credentials and payload data via HTTPS.

The component Enterprise Security and Management (ESM) provides AllegroGraph the mission critical functionality that organizations need to support 24 hours operations, including a transport layer security to and from database clients, which also have the ability to send and receive encrypted requests, and data access control at the triple level security. Moreover, AllegroGraph introduces Triple/Quad Level Security Filters, which can prevent access (both read and write) to triples with a specified value for subject, predicate, object and/or graph, as it is possible to specify which values should be allowed or disallowed. Oracle 12c has the default access control at the graph level that allows the owner of a graph to grant appropriate privileges to other users. However, for applications with stringent security requirements, it is possible to enforce a fine-grained access control mechanism allowing security administrators to define policies that eventually restrict a user's access to triples that involve instances of a specific RDF class or property. Oracle 12c supports also standard encryption that allows administrators to choose which data to encrypt and which standard encryption algorithm to use.

Marmotta has an integrated security module which implements the authentication and authorization mechanism based on the access control list (ACL), that specify which users are granted access to specific objects and what operations are allowed.

Versioning. AllegroGraph, Stardog and Virtuoso do not have a built-in versioning mechanism. In such cases, on top of these stores, external components (e.g. Graph Versioning System) can be deployed in order to keep track of the RDF data contained in the underlying repository. However, this solution is less efficient compared to the case in which the versioning functionality is internal to the RDF store.

Versioning is available in Marmotta on top of the supported backend KiWi triple store. Currently, it allows logging of changes applied to the repository resources, including the source (origin) of the saved triples. Moreover, it offers the capability to create snapshots of the repository that have reached an acceptable level of quality and stability. These snapshots can be later referenced while updating the repository with new data. Instead reverting changes has not yet been implemented and will be added later.

As for many applications the unlimited history is not required, Blazegraph provides a configurable policy to retain historical data for a given period. At any point of this interval, it is possible to request a consistent view of the database and the corresponding data can be read. Most of the stores surveyed do not currently implement the versioning feature in an exhaustive and satisfactory manner. However, much of them are making great efforts to fill this gap, despite the difficulties linked to several constraints imposed on versioning algorithms by RDF model. Oracle 12c is among the technologies that implement the versioning capability in a more comprehensive way, thanks to his component Workspace Manager which provides a mechanism where the granularity (unit) of versioning is a graph model. It supports a collaborative development project where a team can share access to a collection of insertions and updates, allowing also control to the applied operations.

Handling binary data. Most of the current implemented stores do not have an optimized BLOB (Binary Large Object) storage, same like in the most of relational databases. Though Stardog, Marmotta and Blazegraph can store any legal RDF value, they are not best used as a blob stores. AllegroGraph supports CLOB (Character Large Object), useful to handle huge strings, but it cannot directly support BLOB. Virtuoso is capable of storing binary data storage, exploiting the built in WebDAV repository, which can host static and dynamic web content. Finally, Oracle 12c support large volume of data under form of BLOB and it can hold up to 4 GB of data for a single resource, thus ensuring an effective storing of digitized information (e.g. images, audio, video).

A.3 Findings

As a result of this survey it can be said that various implementations of RDF store are suitable to be used as backbone of semantic applications that need to store and process large amount of RDF data in a safe and reliable manner. Moreover, most of these systems usually provide support for essential features such as the capability to guarantee data protection and to backup and restore data. However, the majority of the surveyed tools do not yet support versioning and handling streaming data in an effective way. As from a industrial perspective they embody essential features, their lack is an important gap representing the most pressing technological barrier that researchers and technicians have to overcome in the next future.

Bibliography

ActiveMQ. Online. URL: `{http://activemq.apache.org}`.

ActiveMQ OpenWire. Online. URL: `{http://activemq.apache.org/openwire.htmlg}`.

Alexopoulos, Kosmas et al. (2016). “A concept for context-aware computing in manufacturing: the white goods case.” In: *International Journal of Computer Integrated Manufacturing* 29.8, pp. 839–849.

AlGeddawy, T and H ElMaraghy (2010). “Co-evolution hypotheses and model for manufacturing planning.” In: *CIRP Annals-Manufacturing Technology* 59.1, pp. 445–448.

Ali, Muhammad Intizar et al. (2015). “A Semantic Processing Framework for IoT-Enabled Communication Systems.” In: *International Semantic Web Conference (2)*, pp. 241–258.

Amazon. Online. URL: `{https://aws.amazon.com/iot-platform/}`.

Armbrust, Michael et al. (2010). “A view of cloud computing.” In: *Communications of the ACM* 53.4, pp. 50–58.

Atlas. Online. URL: `{http://www.mclaren.com/appliedtechnologies/products/item/atlas/}`.

Atzori, Luigi, Antonio Iera, and Giacomo Morabito (2010). “The internet of things: A survey.” In: *Computer networks* 54.15, pp. 2787–2805.

Baldoni, Roberto and Luca Montanari (2016). “Italian National Cyber Security Framework.” In: *Proceedings of the International Conference on Security and Management (SAM)*. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), p. 168.

Beckett, Dave and Jeen Broekstra (2008). “SPARQL query results XML format.” In: *W3C Recommendation* 15.

Bellifemine, Fabio Luigi, Giovanni Caire, and Dominic Greenwood (2007). *Developing multi-agent systems with JADE*. Vol. 7. John Wiley & Sons.

Berners-Lee, Tim, James Hendler, Ora Lassila, et al. (2001). “The semantic web.” In: *Scientific american* 284.5, pp. 28–37.

Bizer, Christian, Tom Heath, and Tim Berners-Lee (2009). “Linked data-the story so far.” In: *Semantic services, interoperability and web applications: emerging concepts*, pp. 205–227.

Bizer, Christian and Andreas Schultz (2009). *The berlin sparql benchmark*.

Boncz, Peter et al. (2013). “The linked data benchmark council project.” In: *Datenbank-Spektrum* 13.2, pp. 121–129.

Bossi, Lorenzo, Stefano Braghin, and Alberto Trombetta (2014). “Multidimensional reputation network for service composition in the internet of things.” In: *Services Computing (SCC), 2014 IEEE International Conference on*. IEEE, pp. 685–692.

Brucker, Achim D et al. (2012). “SecureBPMN: Modeling and enforcing access control requirements in business processes.” In: *Proceedings of the 17th ACM symposium on Access Control Models and Technologies*. ACM, pp. 123–126.

Capozzi, Francesco et al. (2014). “Lightweight augmented reality tools for lean procedures in future factories.” In: *International Conference on Augmented and Virtual Reality*. Springer, pp. 232–246.

Cocco, L and P Daponte (2008). “Metrology and formula one car.” In: *Instrumentation and Measurement Technology Conference Proceedings, 2008. IMTC 2008. IEEE*. IEEE, pp. 755–760.

Corsaro, Angelo et al. (2006). “Quality of service in publish/subscribe middleware.” In: *Global Data Management* 19.20, pp. 1–22.

Costa, Daniel L et al. (2014). *An Ontology for Insider Threat Indicators Development and Applications*. Tech. rep. CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST.

Costa, Gianni et al. (2013). “Hierarchical clustering of XML documents focused on structural components.” In: *Data & Knowledge Engineering* 84, pp. 26–46.

Cudré-Mauroux, Philippe et al. (2013). “NoSQL databases for RDF: an empirical evaluation.” In: *International Semantic Web Conference*. Springer, pp. 310–325.

Cybersecurity Framework Manufacturing Profile. Online. URL: {`http://csrc.nist.gov/cyberframework/`}.

EFFRA (2016). *Factories 4.0 and Beyond*. Tech. rep.

Endsley, Mica R and DJ Garland (2000). “Theoretical underpinnings of situation awareness: A critical review.” In: *Situation awareness analysis and measurement*, pp. 3–32.

FIA (2011). *Formula One Technical Regulations. Section 3.18. (2010)*. Tech. rep. Federation Internationale de l’Automobile.

FIPA. IEEE Computer Society. Online. URL: {`http://www.fipa.org/specs/fipa00037/SC00037J.html`}.

Forrester, Jay W (1995). “The beginning of system dynamics.” In: *McKinsey Quarterly*, pp. 4–17.

Fortino, Giancarlo et al. (2013). “An agent-based middleware for cooperating smart objects.” In: *International Conference on Practical Applications of Agents and Multi-Agent Systems*. Springer, pp. 387–398.

Furno, Domenico, Vincenzo Loia, and Mario Veniero (2010). “A fuzzy cognitive situation awareness for airport security.” In: *Control and cybernetics* 39.4, pp. 959–982.

Furno, Domenico et al. (2011). "Towards an agent-based architecture for managing uncertainty in situation awareness." In: *Intelligent Agent (IA), 2011 IEEE Symposium on*. IEEE, pp. 1–6.

Geraci, Anne et al. (1991). *IEEE standard computer dictionary: Compilation of IEEE standard computer glossaries*. IEEE Press.

Guo, Yuanbo, Zhengxiang Pan, and Jeff Heflin (2005). "LUBM: A benchmark for OWL knowledge base systems." In: *Web Semantics: Science, Services and Agents on the World Wide Web* 3.2, pp. 158–182.

Gyrard, Amelie, Christian Bonnet, and Karima Boudaoud (2014). "Enrich machine-to-machine data with semantic web technologies for cross-domain applications." In: *Internet of Things (WF-IoT), 2014 IEEE World Forum on*. IEEE, pp. 559–564.

Gyrard, Amelie et al. (2015). "Cross-domain Internet of Things application development: M3 framework and evaluation." In: *Future Internet of Things and Cloud (FiCloud), 2015 3rd International Conference on*. IEEE, pp. 9–16.

Hankel, M (2016). *Digitising Manufacturing 2016 RAMI 4.0 Reference Architecture Model Industry 4.0*. URL: `http://www.the-mtc.org/Events/Digitising_Manufacturing_2016/4-2_RAMI-Martin-Hankel.pdf`.

Harris, Steve, Andy Seaborne, and Eric Prud'hommeaux (2013). "SPARQL 1.1 query language." In: *W3C recommendation* 21.10.

Hermann, Mario, Tobias Pentek, and Boris Otto (2016). "Design principles for industrie 4.0 scenarios." In: *System Sciences (HICSS), 2016 49th Hawaii International Conference on*. IEEE, pp. 3928–3937.

Hohpe, Gregor and Bobby Woolf (2004). *Enterprise integration patterns: Designing, building, and deploying messaging solutions*. Addison-Wesley Professional.

IDEF. Online. URL: `{http://www.idef.com/idefo-function_modeling_method/}`.

IEEE Foundation for Intelligent Physical Agents (FIPA) Standards Committee. Online. URL: `{http://www.fipa.org/}`.

INTER-IoT. Online. URL: `{http://www.inter-iot-project.eu/open-call}`.

Jürjens, Jan (2005). *Secure systems development with UML*. Springer Science & Business Media.

Kádár, Botond, Walter Terkaj, and Marco Sacco (2013). "Semantic Virtual Factory supporting interoperable modelling and evaluation of production systems." In: *CIRP Annals-Manufacturing Technology* 62.1, pp. 443–446.

Kádár, Botond et al. (2010). "Enhanced control of complex production structures by tight coupling of the digital and the physical worlds." In: *CIRP Annals-Manufacturing Technology* 59.1, pp. 437–440.

Katasonov, Artem et al. (2008). "Smart Semantic Middleware for the Internet of Things." In: *ICINCO-ICSO* 8, pp. 169–178.

Kostelnik, Peter, Martin Sarnovsk, and Karol Furdik (2011). “The semantic middleware for networked embedded systems applied in the internet of things and services domain.” In: *Scalable Computing: Practice and Experience* 12.3, pp. 307–316.

Kuts, Vladimir et al. (2017). “Exploiting Factory Telemetry to Support Virtual Reality Simulation in Robotics Cell.” In: *International Conference on Augmented Reality, Virtual Reality and Computer Graphics*. Springer, pp. 212–221.

Lasi, Heiner et al. (2014). “Industrie 4.0.” In: *Wirtschaftsinformatik* 56.4, pp. 261–264.

Lee, Jay, Behrad Bagheri, and Hung-An Kao (2015). “A cyber-physical systems architecture for industry 4.0-based manufacturing systems.” In: *Manufacturing Letters* 3, pp. 18–23.

Leitão, Paulo (2009). “Agent-based distributed manufacturing control: A state-of-the-art survey.” In: *Engineering Applications of Artificial Intelligence* 22.7, pp. 979–991.

Leitão, Paulo and Francisco Restivo (2006). “ADACOR: A holonic architecture for agile and adaptive manufacturing control.” In: *Computers in industry* 57.2, pp. 121–130.

Leitão, Paulo et al. (2015). “Multiagent system integrating process and quality control in a factory producing laundry washing machines.” In: *IEEE Transactions on Industrial Informatics* 11.4, pp. 879–886.

Leitao, Paulo et al. (2016). “Smart agents in industrial cyber–physical systems.” In: *Proceedings of the IEEE* 104.5, pp. 1086–1101.

Leitner, Stefan-Helmut and Wolfgang Mahnke (2006). “OPC UA–service-oriented architecture for industrial applications.” In: *ABB Corporate Research Center*.

Lemaignan, Severin et al. (2006). “MASON: A proposal for an ontology of manufacturing domain.” In: *Distributed Intelligent Systems: Collective Intelligence and Its Applications, 2006. DIS 2006. IEEE Workshop on*. IEEE, pp. 195–200.

Manco, Giuseppe et al. (2017). “Fault Detection and Explanation through Big Data Analysis on Sensor Streams.” In: *Expert Systems with Applications*.

McAfee, Andrew (2009). *Enterprise 2.0: New collaborative tools for your organization’s toughest challenges*. Harvard Business Press.

McAfee, Andrew, Erik Brynjolfsson, Thomas H Davenport, et al. (2012). “Big data: the management revolution.” In: *Harvard business review* 90.10, pp. 60–68.

McAfee, Andrew P (2006). “Enterprise 2.0: The dawn of emergent collaboration.” In: *MIT Sloan management review* 47.3, p. 21.

Mendes, Marcelo, Pedro Bizarro, and Paulo Marques (2013). “Towards a standard event processing benchmark.” In: *Proceedings of the 4th ACM/SPEC International Conference on Performance Engineering*. ACM, pp. 307–310.

MicrosoftIoT. Online. URL: `{https://www.microsoft.com/en-us/internet-of-things/}`.

Modoni, GE et al. (2017a). “Enhancing factory data integration through the development of an ontology: from the reference models reuse to the semantic conversion of the legacy models.” In: *International Journal of Computer Integrated Manufacturing* 30.10, pp. 1043–1059.

Modoni, Gianfranco E, Marco Sacco, and Walter Terkaj (2014a). “A semantic framework for graph-based enterprise search.” In: *Applied Computer Science* 10.4, pp. 66–74.

— (2014b). “A survey of RDF store solutions.” In: *Engineering, Technology and Innovation (ICE), 2014 International ICE Conference on*. IEEE, pp. 1–7.

— (2016). “A telemetry-driven approach to simulate data-intensive manufacturing processes.” In: *Procedia CIRP* 57, pp. 281–285.

Modoni, Gianfranco E and Davide Tosi (2016). “Correlation of weather and moods of the Italy residents through an analysis of their tweets.” In: *Future Internet of Things and Cloud Workshops (FiCloudW), IEEE International Conference on*. IEEE, pp. 216–219.

Modoni, Gianfranco E, Mario Veniero, and Marco Sacco (2016). “Semantic knowledge management and integration services for AAL.” In: *Italian Forum of Ambient Assisted Living*. Springer, pp. 287–299.

Modoni, Gianfranco E et al. (2017b). “Semantic based events signaling for AAL systems.” In: *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–15.

Modoni, Gianfranco E et al. (2015). “The knowledge reuse in an industrial scenario: A case study.” In: *Eknow 2015, the Seventh International Conference on Information, Process, and Knowledge Management*, pp. 66–71.

Moghaddam, Mohsen and Shimon Y Nof (2017). “The collaborative factory of the future.” In: *International Journal of Computer Integrated Manufacturing* 30.1, pp. 23–43.

Monostori, László (2014). “Cyber-physical production systems: Roots, expectations and R&D challenges.” In: *Procedia CIRP* 17, pp. 9–13.

Moser, Thomas et al. (2009). “Semantic event correlation using ontologies.” In: *OTM Confederated International Conferences” On the Move to Meaningful Internet Systems”*. Springer, pp. 1087–1094.

Mourtzis, Dimitris and Ekaterini Vlachou (2016). “Cloud-based cyber-physical systems and quality of services.” In: *The TQM Journal* 28.5, pp. 704–733.

OPC UA net. Online. URL: `{http://opcfoundation.github.io/UA-.NET/}`.

OPC UA. Online. URL: `{http://opcconnect.opcfoundation.org/2016/09/bringing-the-power-of-the-cloud-to-industrial-automation/}`.

Patti, Edoardo et al. (2016). “Event-driven user-centric middleware for energy-efficient buildings and public spaces.” In: *IEEE Systems Journal* 10.3, pp. 1137–1146.

Project Jersey. Online. URL: `{https://jersey.java.net/}`.

Prud, Eric, Andy Seaborne, et al. (2006). “SPARQL query language for RDF.” In:

Razzaque, Mohammad Abdur et al. (2016). “Middleware for internet of things: a survey.” In: *IEEE Internet of Things Journal* 3.1, pp. 70–95.

RDF. Online. URL: `{https://www.w3.org/RDF/}`.

Robinson, Ian, Jim Webber, and Emil Eifrem (2015). *Graph databases: new opportunities for connected data.* " O'Reilly Media, Inc.".

Rocha, Andre et al. (2014). "An agent based framework to support plug and produce." In: *Industrial Informatics (INDIN), 2014 12th IEEE International Conference on*. IEEE, pp. 504–510.

Rosen, Roland et al. (2015). "About the importance of autonomy and digital twins for the future of manufacturing." In: *IFAC-PapersOnLine* 48.3, pp. 567–572.

Sacco, Marco et al. (2014). "Supporting the design of AAL through a SW integration framework: the D4All project." In: *International Conference on Universal Access in Human-Computer Interaction*. Springer, pp. 75–84.

Sadalage, Pramod J and Martin Fowler (2012). *NoSQL distilled: a brief guide to the emerging world of polyglot persistence*. Pearson Education.

Seaborne, A (2013). "SPARQL 1.1 Query results CSV and TSV formats." In: *W3C Recommendation* 21.

Seaborne, Andy (2011). "SPARQL 1.1 Query Results JSON Format." In: *W3C Recommendation. W3C (21 March 2013). Available at http://www. w3. org/TR/2013/REC-sparql11-results-json-20130321*.

Sheth, Amit, Cory Henson, and Satya S Sahoo (2008). "Semantic sensor web." In: *IEEE Internet computing* 12.4.

Shi, Weisong et al. (2016). "Edge computing: Vision and challenges." In: *IEEE Internet of Things Journal* 3.5, pp. 637–646.

Shostack, Adam (2014). *Threat modeling: Designing for security*. John Wiley & Sons.

SPARQL. Online. URL: `{https://www.w3.org/2009/sparql/wiki/Main_Page}`.

SPARQL Working Group. Online. URL: `{https://www.w3.org/2009/sparql/wiki/Main_Page}`.

Spoladore, Daniele, Gianfranco Modoni, and Marco Sacco (2016). "Smart Devices for the Home of the Future: A New Model for Collaboration." In: *ERCIM NEWS* 104, pp. 48–49.

Stardog. Online. URL: `{http://stardog.com/}`.

Stuckenschmidt, Heiner et al. (2010). "Towards expressive stream reasoning." In: *Dagstuhl Seminar proceedings*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

Terkaj, Walter, Tullio Tolio, and Marcello Urgo (2015). "A virtual factory approach for in situ simulation to support production and maintenance planning." In: *CIRP Annals-Manufacturing Technology* 64.1, pp. 451–454.

Tolio, T et al. (2010). "SPECIES-Co-evolution of products, processes and production systems." In: *CIRP Annals-Manufacturing Technology* 59.2, pp. 672–693.

Uhlmann, Eckart, Eckhard Hohwieler, and Claudio Geisert (2017). "Intelligent production systems in the era of Industrie 4.0–changing mindsets and business models." In: *Journal of Machine Engineering* 17.

Van Brussel, Hendrik et al. (1998). “Reference architecture for holonic manufacturing systems: PROSA.” In: *Computers in industry* 37.3, pp. 255–274.

Vermesan, O and P Friess (2015). *Building the Hyperconnected Society–IoT Research and Innovation Value Chains, Ecosystems and Markets, ISBN: 978-87-93237-99-5*.

Waldo, Jim (2005). “Embedded computing and Formula One racing.” In: *IEEE Pervasive Computing* 4.3, pp. 18–21.

Wang, Shiyong et al. (2016). “Towards smart factory for industry 4.0: a self-organized multi-agent system with big data based feedback and coordination.” In: *Computer Networks* 101, pp. 158–168.

Wells, Lee J et al. (2014). “Cyber-physical security challenges in manufacturing systems.” In: *Manufacturing Letters* 2.2, pp. 74–77.

WINTAX. Online. URL: `{http://www.magnetimarelli.com/business_areas/motorsport/software/wintax4}`.