FIRST – Project Number: 734599

H2020-MSC-RISE-2016 Ref. 6742023

D1.3 Overview of Existing Interoperability of Virtual Factories

Lead Editor: Lai Xu and Paul de Vrieze

With contributions from: John Kasse, Paul de Vrieze, Lai Xu from BU; Federica Mandreoli, Giacomo Gabri from UniMore; Massimo Mechella from SAPIENZA; Alexander Lazovik, Niels Meima from RuG; Yuewei Bai, Guorong Cai, Liu Kai, Xiaogang Wang from SSPU; Fenfang Zeng from KM; Norbert Eder, Stephan Boese from GK; Weiming Gu from Shuangchi.

Deliverable nature	Report
Dissemination level	Public
Contractual delivery date	30/06/2019
Actual delivery date	31/07/2019
Version	V1.0
Keywords	Interoperability, Architecture, FIWARE,
Revision History	BU: Overall, Section 1, Section 2, Section 6, Section 8 and Section 9 RuG: Section 3 UniMore: Section 4 SAPIENZA: Section5 SSPU: Section 6 GK: Section 6 KM: Section 7 Shuangchi: Section 8

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 734599 H2020-MSC-RISE-2016 Ref. 6742023

Table of Contents

Executive Summary

The FIRST (vF Interoperation suppoRting buSiness innovation) project aims to provide new technology and methodologies to describe manufacturing assets; to compose and integrate existing services into collaborative virtual manufacturing processes; and to deal with evolution of changes. This deliverable describes the state of the art of existing interoperability of virtual factory Task T1.6 Inventories and assess existing manufacturing interoperate frameworks and analysis requirements and defining use cases for design and validation of the proposed interoperate framework, thus creating a baseline for the research, in relation to working package WP5.

Section 1 includes a general introduction of the deliverable. Section 2 reviews interoperability of virtual factory architecture in the context of the RAMI 4.0 architecture, international/industrial data space (IDS), interoperability framework for digital manufacturing platforms, FIWARE smart industry architecture, and sources of further information.

Section 3 looks at the issues how to dynamically switch between streaming data sources. Numbers of approaches are reviewed for supporting dynamic on-the-fly switching between streaming data sources for distributed big data analysis and data source interoperability.

Multiple manufacturing domains encompass a wide variety of systems where each of them has their own formats, concepts, relationships, data structures, syntaxes and semantics. Section 4 reviews 10 approaches related to different manufacturing aspects of manufacturing assets/services classification for interoperability of virtual factory. Section 5 provides principles of manufacturing services discovery and composition methods for supporting interoperability of digital factory. Section 6 briefly summarized FIWARE platform and describe how FIRST interoperability framework could be built based on FIWARE platform.

Section 7 introduces KM software manufacturing solutions and provides data interoperability requirements to FIRST framework. Section 8 includes a case study of Shuangchi shoe manufacturing processes and analyses the current process limitations.

The overview of existing interoperability related technologies provides blueprint for our further research on WP5 as well as WP2, WP3, WP4. The initial research in this deliverable is going to be refined in a later release of this deliverable, following the results of WP2, WP3, WP4, and WP5.

1. Introduction

1.1. Purpose and Scope

The main purpose of this deliverable is to set the common grounds for the research to be conducted in FIRST to achieve interoperability of virtual factories. It describes several related technologies such as dynamic on-the-fly switching between streaming data sources for data source interoperability; different semantic description and reasoning approaches among manufacturing assets, manufacturing service, and manufacturing processes; related platforms, e.g. FIWARE platform; IDS, etc.

KM software is a manufacturing enterprise software provider with products relevant to the project. As such, this deliverable also reviews KM software and solutions. Related data interoperability requirements from KM to FIRST are specified. For Shuangchi Industry as an application company, the related shoe manufacturing processes are provided and the limitations of the current processes are identified. Collectively related technologies, platform and requirements are provided in this deliverable.

1.2. Deliverable Structure

The general structure of this deliverable are include two parts. One part includes the related state-of-art technologies. Another part includes description of our industrial partners work and requirements to FIRST interoperability framework.

the RAMI 4.0 architecture, international/industrial data space (IDS), interoperability framework for digital manufacturing platforms are reviewed in Section 2. Section 3 looks at the issues how to dynamically switch between streaming data sources. Ten approaches related to different manufacturing aspects of manufacturing assets/services classification for interoperability of virtual factory are reviewed in Section 4. Principles of manufacturing services discovery and composition methods for supporting interoperability of digital factory are reviewed in Section 5. Section 6 briefly summarized FIWARE platform and describe how FIRST interoperability framework could be built based on FIWARE platform.

KM software manufacturing solutions and related data interoperability requirements to FIRST framework are presented in Section 7. A case study of Shuangchi shoe manufacturing processes and analyses the current process limitations are provided by Shuangchi Industry in Section 8.

1.3. Methodology

The information in this deliverable has been provided by the FIRST workpackage leaders. These inputs are based on literature studies, their knowledge and expertise, as well as that of other WP contributors on the past and on-going projects in the area. When needed and within reason, however, these inputs have been edited by the deliverable editor for the sake of improved readability of the overall deliverable.

2. Interoperability of Virtual Factory Architecture

2.1. Interoperability of Virtual Factory Architecture in the context of the RAMI 4.0 Architecture

- Industry 4.0 requires integration on many levels with many purposes. To bring all this together a common vocabulary is needed.
- RAMI is first of all a framework for thinking about the various efforts that constitute Industry 4.0. It spans the entire product life cycle & value stream axis, hierarchical structure axis and functional classification axis. This allows common understanding and placing of standards in the picture.
- Sees standardization and non-proprietary solutions as essential for industry 4.0
- RAMI4.0 is also formally described as OWL ontology.

RAMI as an architecture itself provides mainly a high-level overview of manufacturing. Overall it is quite comprehensive, and clearly based upon extensive involvement of various manufacturing stakeholders. Very valuable beyond RAMI are the various concepts that it defines. They are key to a deeper conceptual understanding of the manufacturing space.

The core concepts in RAMI of relevance to FIRST are:

Asset (Section 3.3)	"*Object which has value for an organization*" Note that the concept of asset is very broad, but key is that assets have a lifetime and need to be identifiable
Service (Section 3.4)	"*separate scope of functions offered by an entity or organization via interfaces*"
Entity (Section 3.5)	"*uniquely identifiable object which is administered in the information world due to its importance*"
Manifest (Section 3.8)	"*externally accessible, defined set of meta-information on the functional and non-functional properties of the relevant I4.0 component*"
Administration shell (Section 3.12)	"*virtual digital and active representation of an I4.0 component in the I4.0 system*" An administration shell is effectively a digital interface to the asset it represents. They could be provided directly by the "smart" asset, or provided indirectly through another system. Note that multiple interfaces per asset are permitted.
Physical world (Section 4.1)	The world of things that exist in a physical shape
Information world	The information world is the world of information as exists conceptually in an information system. While all this requires physical carriers to hold the information, those carriers are not part of the information world.
Model world	The model world is that of representations of the physical world that would be used to actually perform actions in the physical world
State world	The state world represents current state of assets, preferably directly

	determined of the asset, but possibly determined indirectly.
Archive world	All aspects of the manufacturing information that have historic value, but are no longer current. Note that archival information may still be valuable in predicting future outcomes.

For more detail, RAMI actually delegates to various other established standards, mainly: DIN EN 61360-1, DIN EN 61360-2, IEC/TR 62794, IEC/TS 62832-1.

2.2. International/Industrial Data Space (IDS)

The Industrial Data Space (now International data space) is a virtual data space using standards and common governance models to facilitate the secure exchange and easy linkage of data in business ecosystems (Otto, Boris, et al., 2019). It thereby provides a basis for creating and using smart services and innovative business processes, while at the same time ensuring digital sovereignty of data owners (Otto, Boris, et al., 2019).

The Industrial Data Space initiative was launched in Germany at the end of 2014 by representatives from business, politics, and research (Otto, Boris, et al., 2016). Meanwhile, it is an explicit goal of the initiative to take both the development and use of the platform to a Global level (Otto, Boris, et al., 2016).

Data sovereignty is a central aspect of the International Data Spaces (Otto, Boris, et al., 2019). It can be defined as a natural person's or corporate entity's capability of being entirely self-determined with regard to its data (Otto, Boris, et al., 2019). It is also the base of building virtual factory or building a co-design and co-creation product platform.

In compliance with common system architecture models and standards, the Reference Architecture Model uses a five-layer structure expressing various stakeholders' concerns and viewpoints at different levels of granularity (Otto, Boris, et al., 2016; 2019).

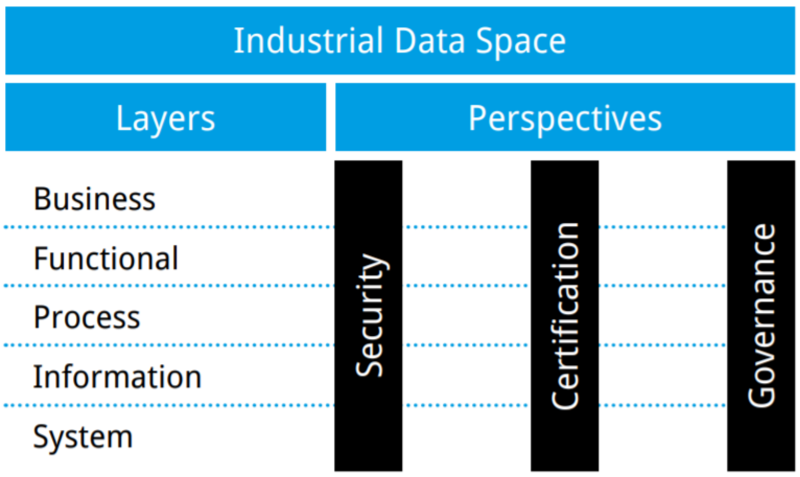


Figure 1. General Structure of Reference Architecture Model (Otto, Boris, et al., 2019)

The general structure of the Reference Architecture Model is illustrated in Figure 1 (Otto, Boris, et al., 2019). The model is made up of five layers: The Business Layer specifies and categorizes the different roles which the participants of the Industrial Data Spaces can assume, and it specifies the main activities and interactions connected with each of these roles (Otto, Boris, et al., 2019). The Functional Layer defines the functional requirements of the International Data Spaces, plus the concrete features to be derived from these (Otto, Boris, et al., 2019). The Process Layer specifies the interactions taking place between the different components of the Industrial Data Spaces; using the BPMN notation, it provides a dynamic view of the Reference Architecture Model (Otto, Boris, et al., 2019). The Information Layer defines a conceptual model which makes use of linked-data principles for describing both the static and the dynamic aspects of the Industrial Data Space's constituents (Otto, Boris, et al., 2019). The System Layer is concerned with the decomposition of the logical software components, considering aspects such as integration, configuration, deployment, and extensibility of these components (Otto, Boris, et al., 2019).

In addition, the Reference Architecture Model comprises three perspectives that need to be implemented across all five layers: Security, Certification, and Governance (Otto, Boris, et al., 2019).

2.3. Interoperability Framework for Digital Manufacturing Platforms

Wajid and Bhullar (2019) reviewed interoperability across digital manufacturing platforms. Many different manufacturing companies intent to build mixed heterogeneous systems by using different platforms, vendor specific technologies, or closed standards. Often the involved platforms and technologies are closed in nature of commercial reasons, which creates interoperability issues particularly concerning cross platform connectivity, utilisation of software applications, and data across multiple platforms.

A platform interoperability framework is proposed in Figure 2 which is designed for interoperability of vertical digital manufacturing platforms. A three tier hierarchy includes the platform tier at the bottom, the application tier at the middle, and the integration tier at the top.

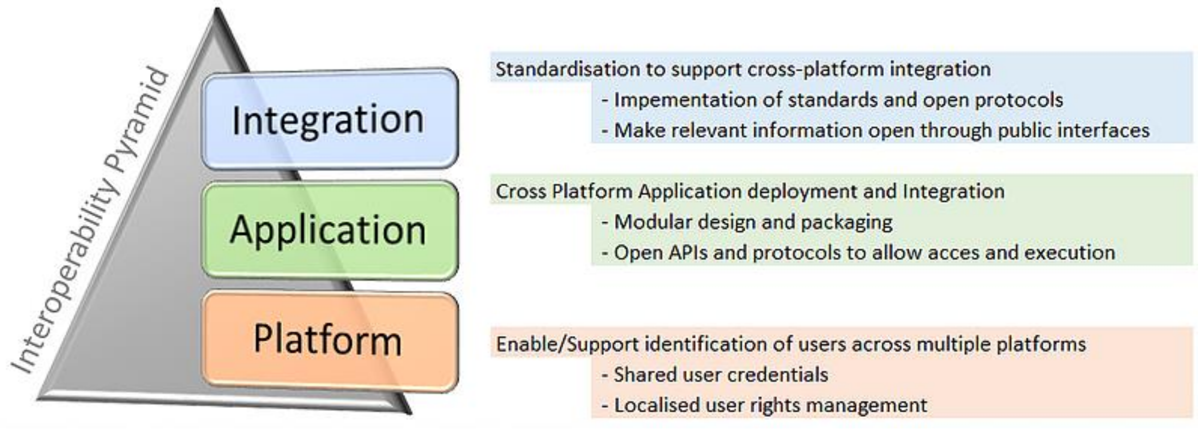

Interoperability Pyramid
Integration
Application
Platform
Standardisation to support cross-platform integration
- Impementation of standards and open protocols
- Make relevant information open through public interfaces
Cross Platform Application deployment and Integration
- Modular design and packaging
- Open APIs and protocols to allow acces and execution
Enable/Support identification of users across multiple platforms
- Shared user credentials
- Localised user rights management

Figure 2. Interoperability Framework for Digital Manufacturing Platforms (Wajid and Bhullar, 2019)

The platform tier focuses on the separation of 'identification' from 'services' to allow shared access across different platforms. Approaches for single sign-on, policy based access and user right management can all contribute towards interoperability at this level (Wajid and Bhullar, 2019). The application tier should focuses on to sharing the application and services with users with right to access the platforms, either locally or through remote access (Wajid and Bhullar, 2019). The integration tier can dealing with heterogeneous standards, interfaces and communication protocols. The use of standards at all stages of the information/data flow can allow the applications, tools and services to be interoperable in an ecosystem environment (Wajid and Bhullar, 2019).

In general, the platform tier could be control by the shared access across different platform, which is not always could be supported by virtual factory management in the FIRST project. The applications and services at the application tier are described as manufacturing assets in the FIRST project. The services at the integration tier can be treated as the manufacturing services or the main services at the interoperability framework in the FIRST project.

2.4. FIWARE SMART industry Architecture

Open-sourced platform, FIWARE (2018) have constructed an architectural model in Figure 3. At shop floor level, there are various machines and systems that will collect data to be processed by the IoT agents, RTPs and System adapters. FIWARE uses its own context broker known as Orion. This is a software component that can be applied to any SMART solution and allows data producers to submit context information such as metadata in a decentralized way. The consumers can then query and retrieve context information from the Orion Broker (CEF Digital, 2019). Large scale big data processing engines such as Hadoop are then used along with business intelligence platforms to enable key performance indicator monitoring and for algorithms to be performed on the data sets. At the top right, FIWARE could also access IDS through IDS connector for the data required from the third organisations in the system.

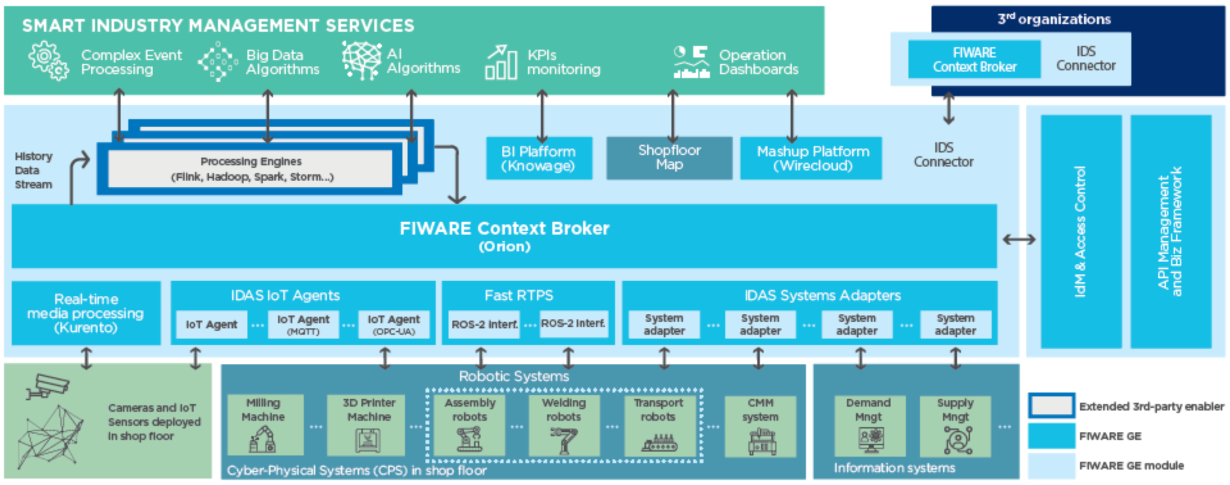


Figure 3. FIWARE SMART industry Architecture (FIWARE, 2018)

2.5. Sources of further information

- Industry 4.0 standards website: http://i40.semantic-interoperability.org/
- Github details: https://i40-tools.github.io/StandardOntologyVisualization/index.html
- “official” website: https://www.zvei.org/en/subjects/industrie-4-0/the-reference-architectural-model-rami-40-and-the-industrie-40-component/
- DIN SPEC 91345:2016-04

3. Dynamic on-the-fly switching between streaming data sources for distributed big data analysis and data source interoperability

The number of data producing devices, continues to keep growing each year. Due to the growth of such devices, the amount of data being gathered also grows (Chen, Mao, and Liu, 2014), (Hashem et al. 2015). The data that is produced has to be analysed, since it might provide valuable insights (Assunção, 2015)(Provost and Fawcett, 2013).

Analysing the vast amounts of data on a single computer is not possible anymore, due to long processing times. Instead, distributed data processing frameworks have been developed which provide the necessary tools and abstractions to make it easier for developers to fulfil their data processing needs in a distributed fashion. Distributed data processing frameworks have reduced the processing times significantly, since data processing is now done in parallel instead of on a single computer (Kambatla et al. 2014). The reduction in processing time might prove valuable to companies that require obtaining possibly valuable insights extracted from the gathered data. The distributed data processing frameworks give developers the tools to write data processing scenarios, in which is described how to retrieve and process data, after which the distributed data processing frameworks takes care of efficient distributed data retrieval and processing.

Making changes to a running processing scenario might often be necessary, especially when it comes to the data sources from which the processing scenario retrieves data to process. For example, due to external changes with respect to the processing scenario, the data sources from which data is retrieved might need to be re-configured. When talking about sensors as data sources it quickly becomes clear that dynamic reconfiguration functionalities are very useful: new and more accurate sensors might become available as data sources, which have to be added to the processing scenario, after which older less accurate sensors might need to be removed as data sources, since they are obsolete.

However, in current distributed data processing frameworks once a processing scenario is started, changes can only be introduced by stopping and restarting the processing scenario. This is unacceptable in environments where the results of the processing analysis are expected in real-time or when the processing of data simply cannot be stopped due to its business critical nature. Therefore, the need for the functionality of dynamically switching data sources becomes clear.

Extensions in the form of libraries have been developed to support dynamic re-configuration functionalities not natively supported by the currently available distributed data processing frameworks. However, such solutions require the use of said frameworks and not all frameworks support such an extension. A generic approach for supporting the dynamic reconfiguration for streaming data sources would be very valuable, not imposing any requirements on the choice of distributed data processing framework.

This research presents an approach which aims to make it possible to dynamically manage streaming data sources without the need of restarting a processing scenario, whilst being truly generic by not posing any requirements on the choice of existing distributed data processing frameworks or streaming data source, hence providing a generic and interoperable solution for modern data science.

3.1. Problem statement

Current distributed data processing frameworks lack the ability to switch between streaming data sources dynamically. Once the processing scenario has been started, changes in streaming data sources can only be introduced by stopping and restarting the processing scenario. The process of stopping and restarting causes downtime with respect to processing, during which no insight into the data can be provided. To solve this issue, the following problems are defined:

RQ1: *Is there a feasible approach to dynamically switching between streaming data sources without restarting the processing scenario?*

RQ2: *What is the most suitable approach to dynamically switching between streaming data sources in terms of architecture and tools?*

Since there is currently no system available which solves the aforementioned problems in the context of distributed data processing, a new system needs to be built. Based on the stated problems, a set of requirements is introduced which the new system needs to fulfill to be regarded as a feasible solution. The set of requirements can be found in Table 1.

Table 1. Requirements which have to be met to realize a solution which allows for dynamically switching streaming data sources

Requirement	Description
Dynamic	The system should allow dynamically switching between streaming data sources without the need of restarting the processing scenario
Generic	The system should be generic in that no requirements are posed with regards to using certain data sources or certain distributed data processing framework
Scaleable	The system should be scalable and run in a distributed fashion
Versioned	The system should have a mechanism in place for versioning data source switches and integrations, since dynamically switching data sources influences the result of and might cause incompatibilities with the processing scenario
Provenanced	The system should have a mechanism in place for providing provenance about the data being integrated into the system, so the realisation of results of a processing scenario can be reasoned about
Extensible / interoperable	The system should be extensible both in that new types of data sources and new types of distributed data processing frameworks can be integrated into the system without making core modifications to the system
Usable	The system should not require significant changes on the side of the processing scenario to be compatible with the system

3.2. Approaches

Distributed data processing frameworks support two kinds of processing types with respect to data sources: batch data processing and streaming data processing. In this paper the main focus will be on dynamic reconfiguration of streaming data sources, since the number of IoT devices which produces streams of events grows steadily without solutions for dynamic re-configuration, as opposed to batch data source for which extensions for existing distributed data processing solutions already exist, albeit in a non-generic form.

There are several approaches for realizing dynamic reconfiguration of streaming data sources. One approach would be to extend the core functionality of existing distributed data processing frameworks. For example, an extension to Apache Spark (2019), called dynamic-spark (Lazovik et al., 2017) has been developed which follows this approach. Although the approach works, it is specific to Apache Spark and its framework native concepts, meaning that extensions for other popular distributed data processing frameworks like Apache Flink (2019), Apache Storm (2019), Apache Beam (2019), Apache Hadoop MapReduce (2019) have to be developed and maintained separately. Therefore, another approach is proposed which is independent of the choice of distributed data processing framework, providing a more generic solution. A set of requirements is defined which should be met by the proposed solution.

Related literature is examined, to see how current solutions meet the set of requirements. Also, a thorough examination of the available tools in the area of distributed computing will be presented. Due to rapid innovations of tools in this area, there is a lack of discussion of the applicability and usefulness of said tools in recent literature with respect to realising for example the proposed solution.

An implementation of the proposed solution is provided. The solution aims to integrate several existing tools, which should provide a solid basis for the solution. Kubernetes (2019), a platform for orchestrating distributed applications, will provide solid primitives for running our solution in a distributed fashion, whilst also allowing for native extensions to be developed through a recent development known as operators (CoreOS, 2019). For realizing a generic and scale-able interface from which any currently popular distributed data processing and futures ones can retrieve data, we will rely on Apache Kafka (2019) which has proven to be most successful distributed streaming platform. For integrating streaming data sources we will use Apache Camel (2019), the most popular integration framework with support for a vast amount of heterogeneous data sources, under which streaming data sources. The tools will be combined with the required business logic to offer the functionality of dynamically switching data sources.

The achieved solution is analysed and its strengths and weaknesses will be discussed, which will finally lead to the proposal of opportunities for future work.

3.3. Architecture

Supporting dynamic data source switching functionality needs several problems to be solved, whilst accounting for the set of requirements posed in Table 1. Through a discussion of the set of requirements an architecture for the system is proposed.

First, the system should be dynamic in terms that it should be possible to dynamically switch between data sources without restarting the processing scenario. Next, the system should be generic in the sense that no requirements are posed with regards to which streaming data sources and which distributed data processing framework should be used. Also, the system should provide a high level of usability, such that users are not expected to make significant changes to their processing scenario to integrate with the system. To fulfill these requirements, an additional layer between the processing scenario and the data sources should be realized. By inserting this additional layer, the control over how streaming data sources are integrated and made available to the processing scenario is now lies with this additional layer. The additional layer will be called the Dynamic Data Provider from now on. The position of the Dynamic Data Provider between processing scenario's and data sources can be seen in Figure 4.

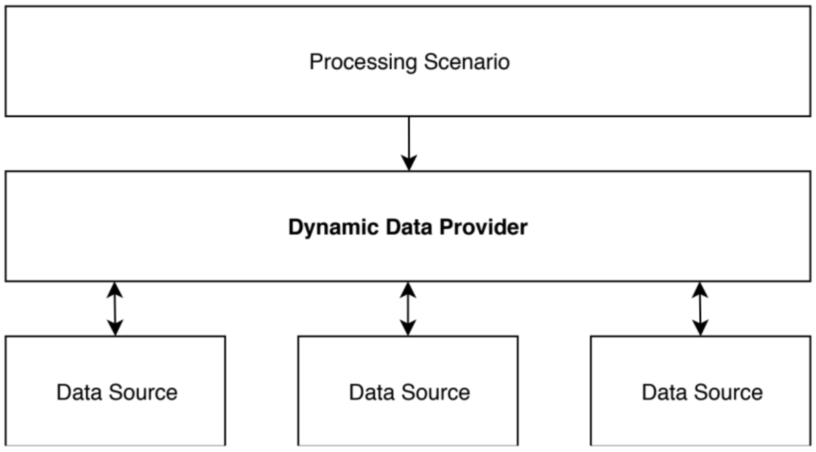


Figure 4. A high-level overview of how the Dynamic Data Provider acts as an additional layer between the processing scenario's and the data sources

The Dynamic Data Provider provides a generic interface for describing how heterogeneous data sources should be integrated. After data sources have been integrated, the Dynamic Data Provider then provides a generic interface to which processing scenarios can connect to retrieve the integrated data. By using the Dynamic Data Provider it becomes possible to apply business logic to how data sources are being integrated and made available to processing scenarios. Also, by introducing such a Dynamic Data Provider such business does not have to be embedded into the processing scenario. By constructing the Dynamic Data Provider, it becomes possible to fulfil the dynamic, generic, extensibility and usability requirements.

In short, the research presented in this section has been focused on the feasibility of dynamically switching between streaming data sources in the context of distributed data processing. By posing a set of requirements, architecture was proposed which lead to an implementation. The implementation was discussed based on whether the requirements were fulfilled.

We conclude by answering research questions which were posed at the beginning of this section. There is indeed a feasible approach to dynamically switching between streaming data sources. By giving a thorough reviewing literature and tools, together with defining a set of requirements, the most suitable approach was determined and converted into architecture. By using Kubernetes as a basis, which allows for a high level of extensibility and interoperability through the Operator Pattern, in combination with Apache Camel, Apache Kafka and the custom business logic required to fulfill the versioning and provenance requirements, a very suitable approach is achieved. Although the feasibility of the system is shown, there are still many opportunities for future work.

4. Manufacturing Assets/Services Classification for Interoperability of Virtual Factory

Multiple manufacturing domains encompass a wide variety of systems where each of them has their own formats, concepts, relationships, data structures, syntaxes and semantics. This brings in complexity and limits interoperability across domains. While they may perform flawlessly within their standalone applications, they do not fare well when there is a need to share knowledge. This identifies the need to address interoperability problems and more specifically minimize the semantic mismatches.

These issues have been historically tackled in the data management research area where the main addressed problem is to effectively manage heterogeneity in a dynamic context while preserving the autonomy of the data sources. Given a collection of disparate and distributed data sources, two main approaches exist: either to provide a unified view (i.e., data integration) or to enable the exchange of data among them (i.e., data exchange) (De Giacomo, Lembo, Lenzerini, Rosati, 2007). In this context, the main difficulty lies in the fact that there is no agreement on the adopted data management systems, data models and languages, the vocabularies and structures used to describe the data (often denoted as schema) and the semantics of data values. Relationships between the data exposed by heterogeneous information systems are usually expressed through mappings that are declarative specifications spelling out the relationship between a target data instance and possibly more than one source data instance (Kolaitis, 2018). The most interesting proposals that can be exploited to support interoperability of virtual factory are dataspaces (Franklin, Halevy, Maier, 2005), peer data management systems (Cudre ́-Mauroux, M. T. Ozsu, 2009), and polystores (Gadepally et al., 2016), (Wang et al., 2017).

However, interoperability of virtual factories exhibits peculiar characteristics, and in the literature we can find specific approaches that aim at tackling those issues. The paper (Szejka, Canciglieri, Panetto, Loures, & Aubry, 2017) provides a systematic literature review to identify the main researches and the milestones reference works on semantic interoperability when industries need to effectively share heterogeneous information during Product Development Process (PDP) within and across their institutional boundaries. Although there are multi-perspectives to PDP, this research considers the semantic interoperability issues between the product domains and the different phases of PDP since they naturally overlap each other and are related to the same product. After a careful evaluation of more than 3600 scientific articles with high impact factor, the analysis and categorisation pointed out 14 articles and 8 authors that were the most relevant researches and the milestones references in the studied area, presenting the knowledge boundaries of this field and offering opportunities to extend it. The authors of this study conclude that key scientific topics to be explored in this research field are: (i) heterogeneous information from multiple domains that should be concurrently related to Product Design and Manufacturing may be formalised in a rigorously defined set of shareable core concepts in an ontological approach and applicable in a semantically interoperable manner; and (ii) heterogeneous information relationships from multiple domains concerning Product Design and Manufacturing may be simultaneously mapped via sets of interoperable mechanisms (semantic rules) for sharing, converting and translating information.

As a matter of fact, different works adopt ontological approach as a common ground for the formal specification of the involved assets and services. Inspired by (Szejka, Canciglieri, Panetto, Loures, & Aubry, 2017) we report the 10 most relevant approaches in the literature.

Table 2. Comparison of the approaches

Approach	**Main ontological approach**	**Ontology construction**	**Ontology exploitation**	**Reasoning**
Panetto 2012	Knowledge-driven ontology merging	Merging of the STEP Standard (ISO 10303) and IEC 62264		First Order Logic
Lu 2014	Knowledge-driven ontology merging	Merging of ISO ontologies	Manufacturing resources virtualization and resource retrieval	
Saha 2017	Ontology specification	Proposal of a Product Lifecycle Ontology (PLO)		Formal logic
Chungoora 2013	Model-driven ontology construction	Proposal of the Manufacturing Core Ontology		Integrity checking
Szejka 2016	Ontology-based Interoperable Product Design and Manufacturing System		Semantic reconciliation and mapping of existing ontologies	
Liao 2014	Semantic annotations of PLM models		Semantic annotation Creation of reasoning rules	
Belkadi 2012	Model-driven meta-model specification for ontology integration		The meta-model is exploited as ontology	
Canciglieri 2010	Mapping method for domain ontologies			
Chen 2010	Knowledge integration and sharing for collaborative moulding product design and process development	Proposal of an IDEF0-based ontology	Ontology-based knowledge integration and mapping between the global ontology and the local ones	

Kim 2006	ontology-based assembly design	Assembly Relational Model (ARM) enhancement		SWRL/OWL-based reasoning

Table 2 summarizes the features of the approaches and compare them. The adopted criteria for the comparison are the following: (1) *Main ontological approach*: brief description of the main contribution related to the adoption of ontologies; (2-4) the other criteria relate to the ontological issues addressed in the corresponding work and concerns three main aspects: ontology *construction*, ontology *exploitation* and *reasoning*. If the surveyed paper deals with the corresponding issue, a brief description of the adopted approach is provided, it is empty otherwise. We briefly describe the 10 approaches; we remark that all could be suitable to be adopted in the FIRST project and in related researches.

Panetto, Dassisti, & Tursi, (2012) proposes a methodology for a knowledge representation process to feed concepts into an ontological model. The methodology is made up of two main steps: the first one consists in conceptualising existing standards, related to product technical data modelling for the definition of products information, providing a ‘‘product-centric’’ information model to represent knowledge and concepts. These concepts can be processed by several enterprise applications within a manufacturing environment: the standards considered for this purpose were the STEP Standard (ISO 10303) and IEC 62264. The first step is based on a syntactical analysis, to compare the instances defined in both standard models and then comparing properties of the shared objects, based on semantic analysis. In this step, mappings between the involved standards were manually designed and then verified through the formalization in First Order Logic (FOL). Taking into account the previous FOL axioms of standard models and the concepts mapping between them, the second step is then to formalize this proposed ‘‘product-centric’’ information model in terms of a Product Ontology.

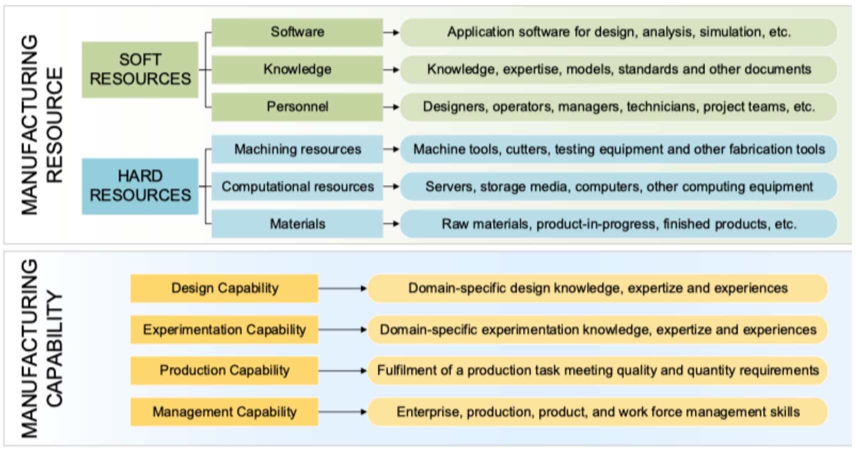

MANUFACTURING RESOURCE
SOFT RESOURCES
Software
Application software for design, analysis, simulation, etc.
Knowledge
Knowledge, expertise, models, standards and other documents
Personnel
Designers, operators, managers, technicians, project teams, etc.
HARD RESOURCES
Machining resources
Machine tools, cutters, testing equipment and other fabrication tools
Computational resources
Servers, storage media, computers, other computing equipment
Materials
Raw materials, product-in-progress, finished products, etc.
MANUFACTURING CAPABILITY
Design Capability
Domain-specific design knowledge, expertize and experiences
Experimentation Capability
Domain-specific experimentation knowledge, expertize and experiences
Production Capability
Fulfilment of a production task meeting quality and quantity requirements
Management Capability
Enterprise, production, product, and work force management skills

Figure 5. Classification of manufacturing resources (Lu, Qun, Chirpreet, Xu, & Ye, 2014)

Lu et al. (2014) presents an ontology-based approach to enabling sematic interoperability throughout the whole process of service provision in the cloud. Indeed, cloud is changing the way enterprises do their businesses in that dynamically scalable and virtualised resources are provided as consumable services over the internet. Consumers can request services ranging from product design, manufacturing, testing, management and all other stages of a product life cycle. Service providers can come together to form a temporary alliance to take manufacturing jobs. In terms of information and knowledge sharing, the most straightforward approach is to use the STEP standard (ISO10303). The paper proposes ontology for cloud manufacturing that incorporates some of the parts of

STEP/STEP-NC. The proposed classification of manufacturing resources is shown in Figure . The paper also discusses how the proposed ontology facilitates manufacturing resources virtualization and resource retrieval.

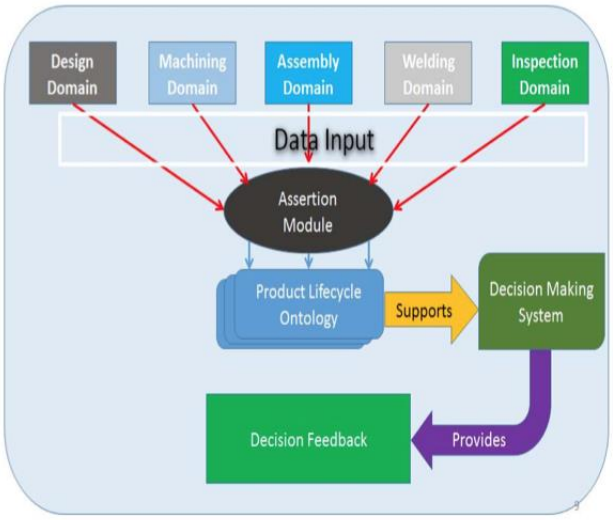


Figure 6. The IPDMS architecture (Saha, Usman, Jones, Kshirsagar, & Li, 2017) overall framework solution

Saha et al. (2017) proposes a formal ontology to support interoperability across multiple product lifecycle domains. A formal ontology is computer interpretable through the use of constraints for restricting the meaning of the terms and has the capability to deduce new knowledge from existing knowledge using inference rules. The proposed formal ontology, Product Lifecycle Ontology (PLO), provides the base for knowledge sharing across design, machining, assembly, welding, and inspection domains. The overall working framework is shown in Figure 6. Data entries are fed through the data assertion module into the PLO. Based on the user's requirements embedded through formal logic and the populated knowledge, PLO makes inferences and provides feedback to support decision making.

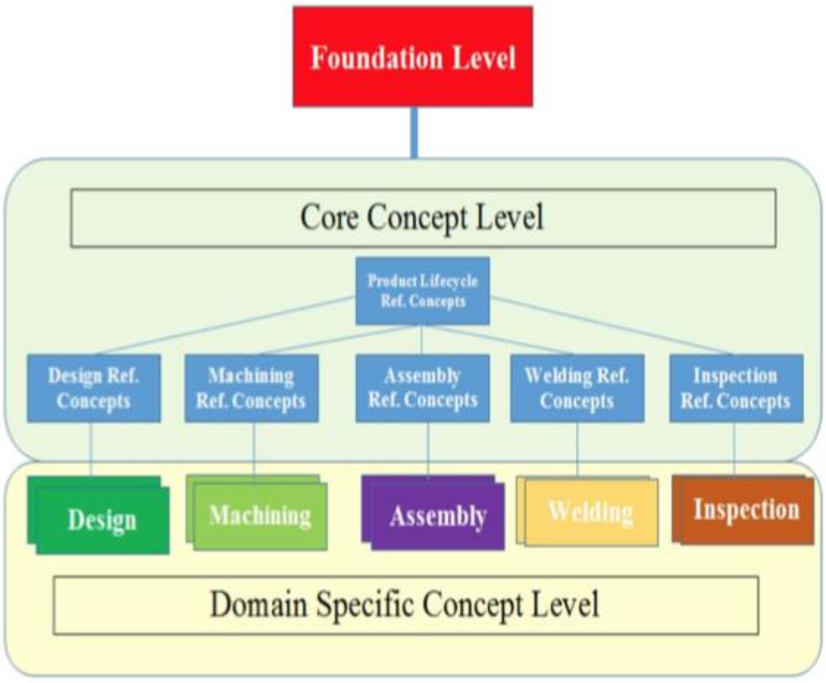


Figure 7. PLO specialization levels (Saha et al., 2017)

The proposed solution PLO uses a layered approach for the different kinds of concepts as shown in Figure 7 at the foundation level concepts are generic enough to be applicable for all kinds of products and across all the stages of the product lifecycle e.g. Event, activity, quantity, object, etc.

This level is specialized into a core concepts level comprising of core manufacturing concepts acting as reference concepts for multiple product lifecycle domains e.g. Product, component, part, dimensions etc. The final level of specialization is the domain specific level comprising of concepts which are more specific for particular domain of design, machining, assembly, welding and inspection such as inspection feature, drilling feature etc.

Chungoora et al. (2013) addresses interoperability of semantics and knowledge in Product Lifecycle Management (PLM), in the direction towards knowledge-driven decision support in the manufacturing industry. The proposed approach identifies a concept based on Model Driven Architecture (MDA), Model Driven Interoperability (MDI) and ontology-driven specifications in order to achieve manufacturing system interoperability and knowledge sharing across multiple platforms in the product lifecycle. They rely on the expressive power of the Extended Common Logic Interchange Format (ECLIF) as ontology language. The proposed approach is part of the Interoperable Manufacturing Knowledge Systems (IMKS), whose architecture is reported in Figure 8. The system relies on three models: the Computation Independent Model (CIM), the Platform Independent Model (PIM) and the Platform Specific Model (PSM). The MDA implies that the models are transformed from the high level (requirements) to the low level (implementation). One of such transformation is from a UML class diagram to ECLIF expression, exploiting the already mention ontology.

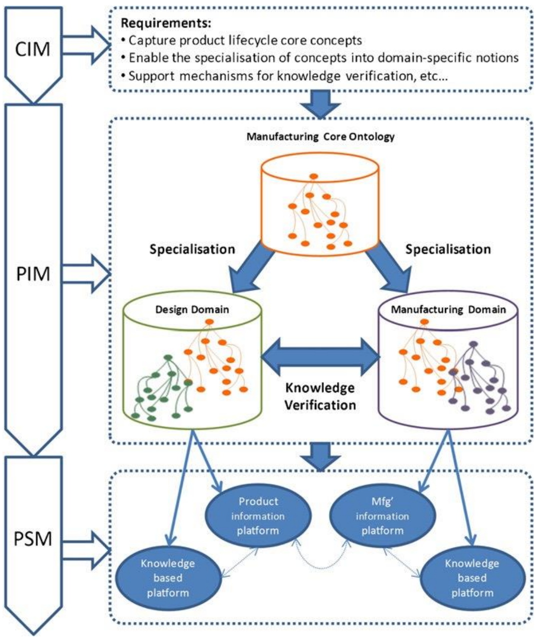

CIM
Requirements:
• Capture product lifecycle core concepts
• Enable the specialisation of concepts into domain-specific notions
• Support mechanisms for knowledge verification, etc...
Manufacturing Core Ontology
Specialisation
Specialisation
PIM
Design Domain
Manufacturing Domain
Knowledge Verification
PSM
Product information platform
Mfg' information platform
Knowledge based platform
Knowledge based platform

Figure 8. The IMKS model-driven concept (source: (Chungoora et al., 2013))

Szejka et al., (2016), the same authors of the previously-mentioned survey, proposed an Interoperable Product Design and Manufacturing System (IPDMS) concept based on a set of engineering domain ontologies and sematic mapping approaches, in order to support semantic interoperability of information in the different phases of the Product Development Process (PDP). They adopt a model-driven approach. The application view is used to support the information sharing between product design and manufacturing and verify the accordance with product requirements. Figure 9 reports the architecture of the IPDMS, which is composed of four views: (i) Foundation View; (ii) Application View; (iii) Semantic Reconciliation View; and (iv) Constraints

View. The three phases of PDP are exposed on the box A in the figure: (i) Conceptual Design Phase; (ii) Tooling Design Phase; and (iii) Manufacturing Design Phase. IPDMS works based on formal models and semantic reconciliation trough the semantic mapping concepts.

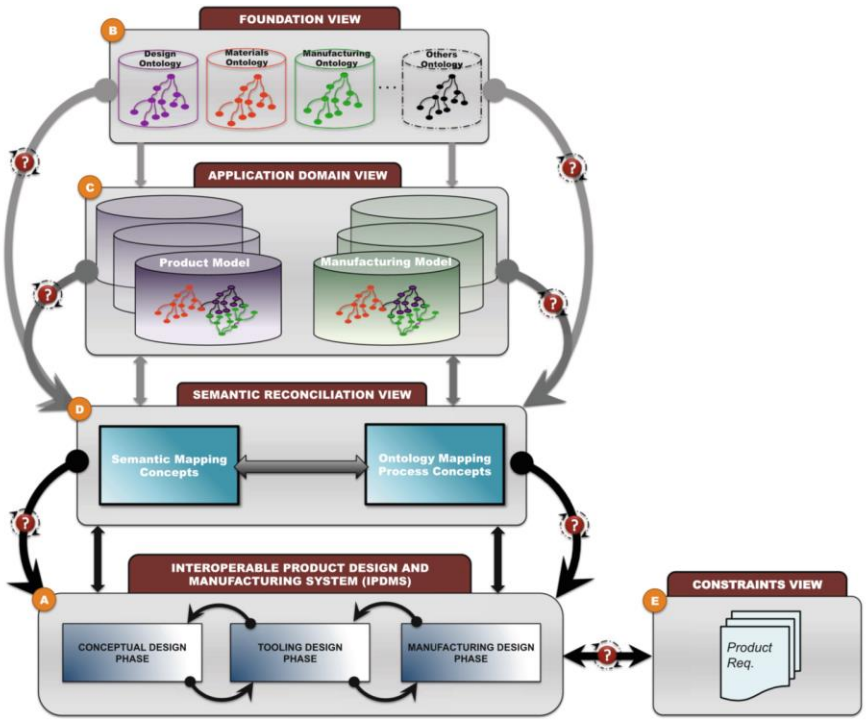


Figure 9. The IPDMS architecture (Szejka et al., 2016)

Liao et al., (2014) proposes a formalisation of semantic annotation for system interoperability from the view of different domains in a Product Life Cycle Management environment. The formalisation made explicit the tacit knowledge in application models and provided support for all activities during the product life cycle. Semantic links are established with different domains and potentially contributes to semantic interoperability across PDP, even though this approach did not depict the information interoperability across PDP. Figure 10 reports the meta-model of the semantic annotation exploited by this approach.

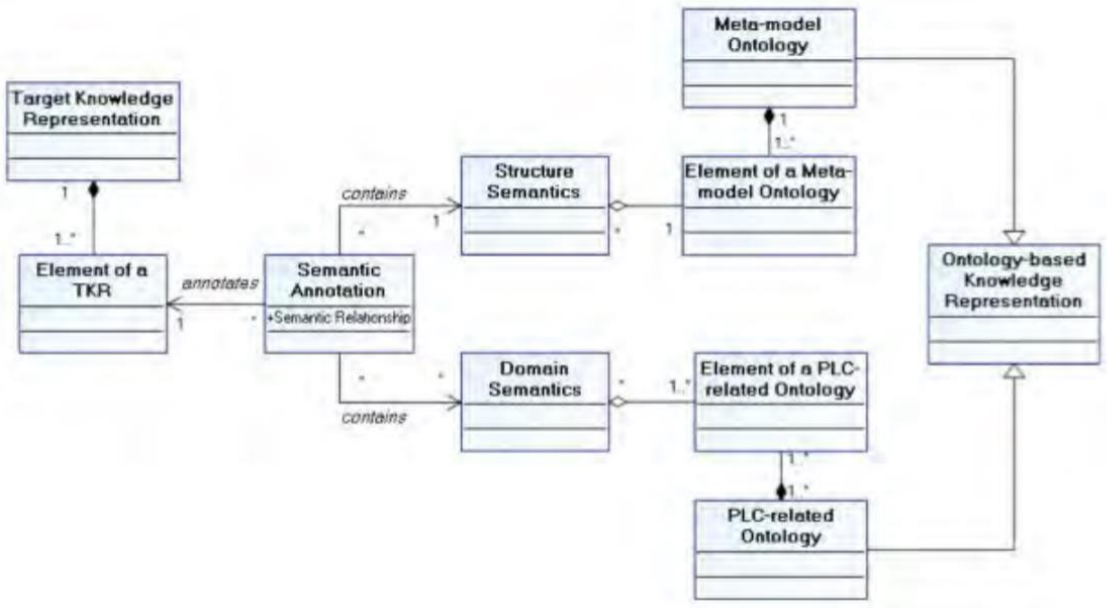


Figure 10. The meta-model of the semantic annotation (Liao et al., 2014)

Belkadi et al., (2012) proposed an approach based on a meta-model to manage the integration of heterogeneous knowledge models of field experts in a process that aims at being collaborative. In Figure 11, Figure 12 and Figure 13 the meta-models exploited in the approach are reported, respectively the core, the data and the collaboration meta-models. Such a distinction represents the differences between the core concepts of knowledge and additional elements. It represents the relation between these concepts and between concepts of heterogeneous experts' models. The proposed approach leverages the communication between different tools. One drawback of the approach is that it does not consider information interoperability between different processes; however, the approach works with core concepts and semantic mapping that are used to support the heterogeneity of information between models, which currently occurs during a collaborative project.

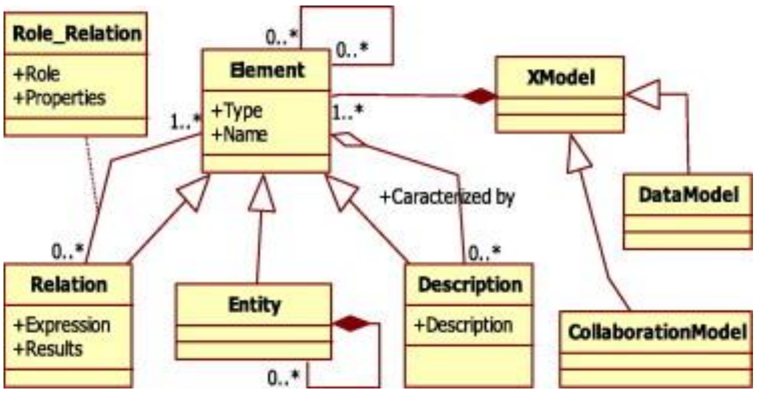


Figure 11. The Meta-Model Core (MMC) (Belkadi et al., 2012)

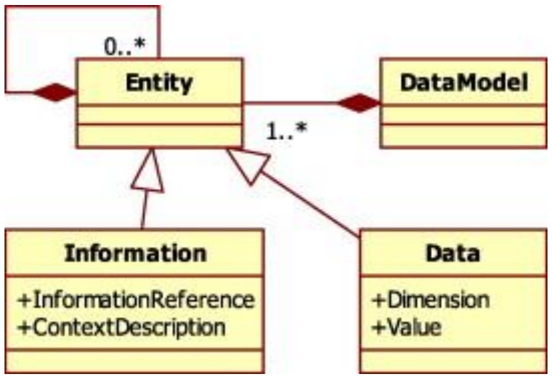


Figure 12. The Data Meta-Model (DMM) (Belkadi et al., 2012).

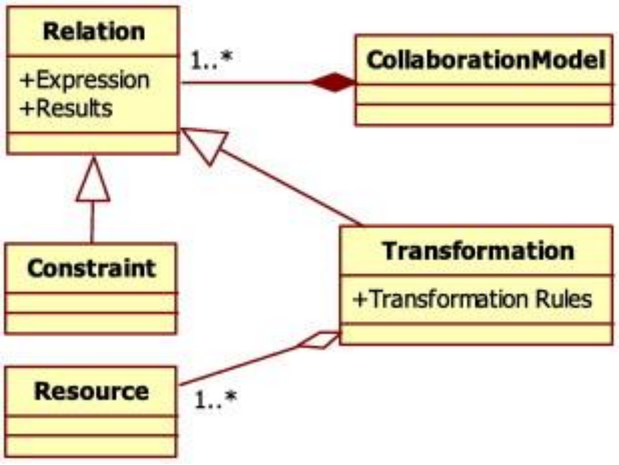


Figure 13. The Collaboration Meta-Model (CMM) (Belkadi et al., 2012).

Canciglieri and Young, (2010) proposed a model-driven approach based on conceptual multiple views, an object-oriented model and UML. They aim at mapping information relationships between designs and manufacturing domains based on translation mechanisms. Each mechanism had a specific knowledge, which was responsible for translating the information from one view to another. Even if the proposed approach is interesting for interoperability, it seems to be limited to specific domains; however, the information structure and the translation mechanisms could be applied to interoperability in product design and manufacturing.

Chen Y. J., (2010) presented an approach to develop knowledge integration and sharing mechanism for collaborative product design and process development. The approach is focused on moulding manufacturing. It includes three steps: the first is the modelling of collaborative moulding product design and process development process; the second aims at defining an ontology-based knowledge model, reported in Figure 14; the third is the design, development and implementation of a framework for knowledge integration and sharing, see Figure 15. The result of this approach significantly contributes to the semantic interoperability in product design and manufacturing framework based on ontological approach.

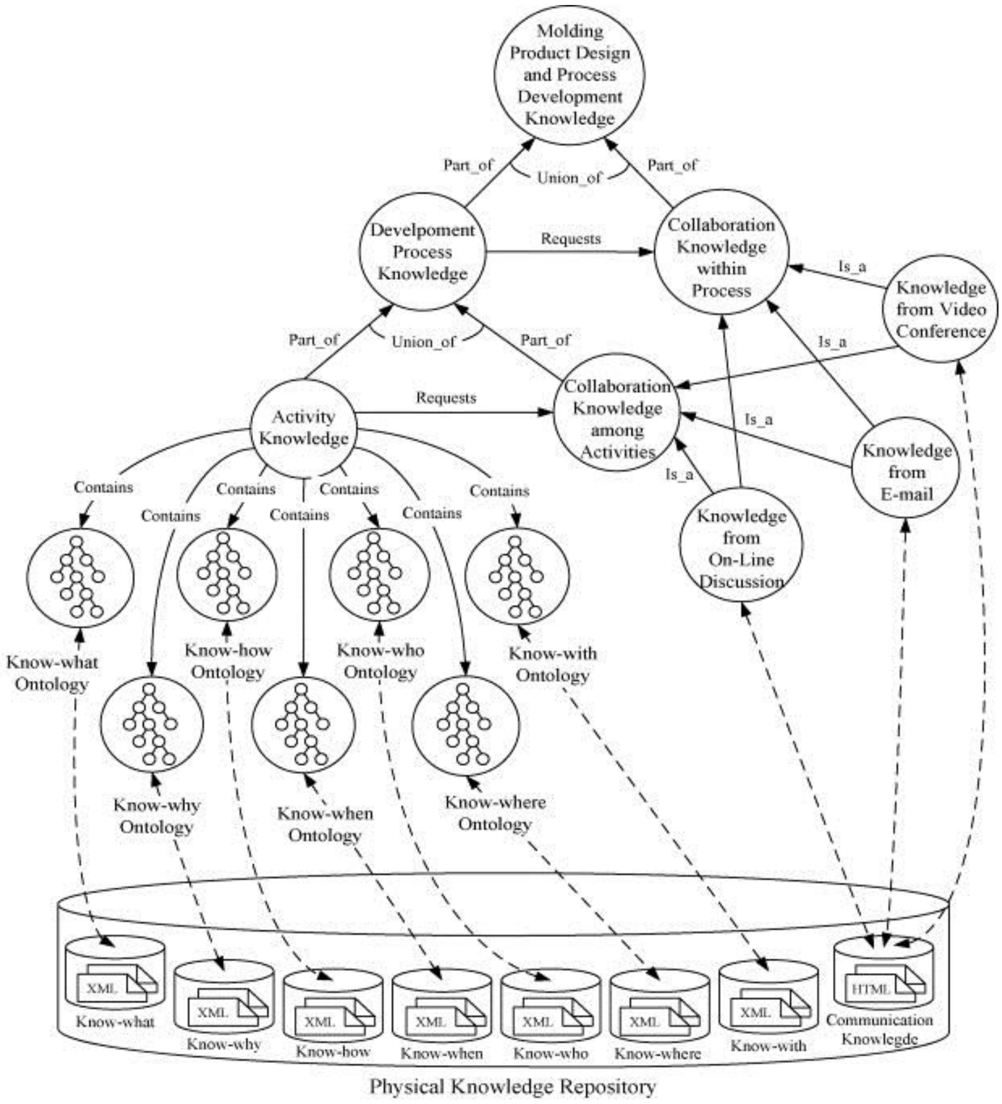


Figure 14. Ontology-based knowledge model (Chen, Y. J., 2010)

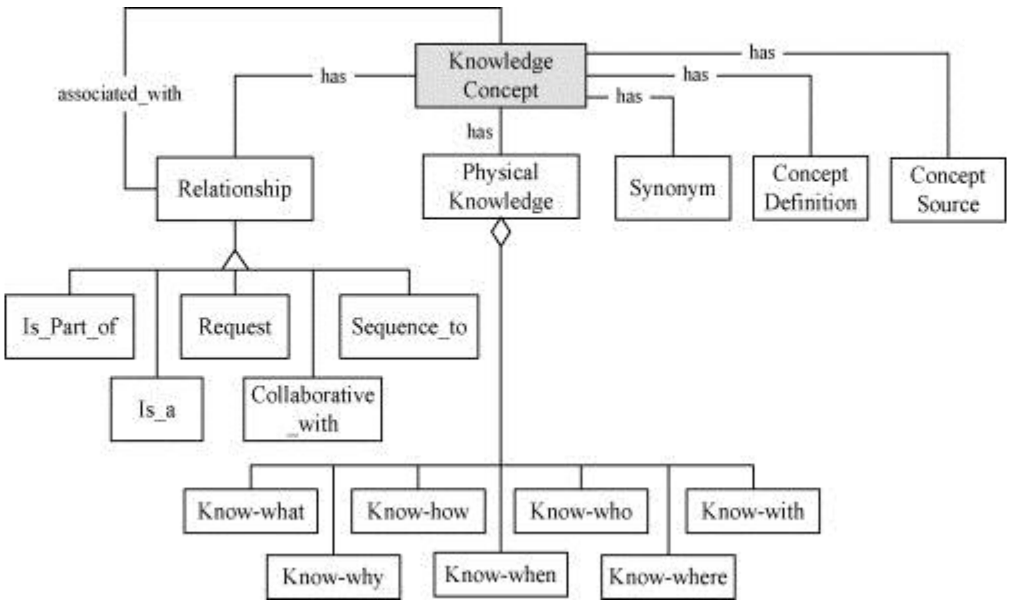


Figure 15. Schema of knowledge concept (ontology) (Chen, Y. J., 2010)

Kim, Manley, Yang, (2006) proposed a specific ontology for assembly design (AsD), reported in Figure 16. Such an ontology is a formal and explicit specification of AsD, with the twofold aim of making assembly knowledge machine-interpretable and of sharing the knowledge itself. An Assembly Relation Model (ARM) is defined, exploiting ontologies that represent engineering, spatial, assembly and joining relations of assembly, leveraging the information sharing and the collaboration in assembly environments. In the proposed AsD ontology, implicit AsD constraints are explicitly represented using OWL (Web Ontology Language) and SWRL (Semantic Web Rule Language). Even if this approach was limited to the assembly domain, the integration of OWL and SWRL is interesting and it can be potentially adopted in Product Design and Manufacturing to overcome the semantic interoperability issues.

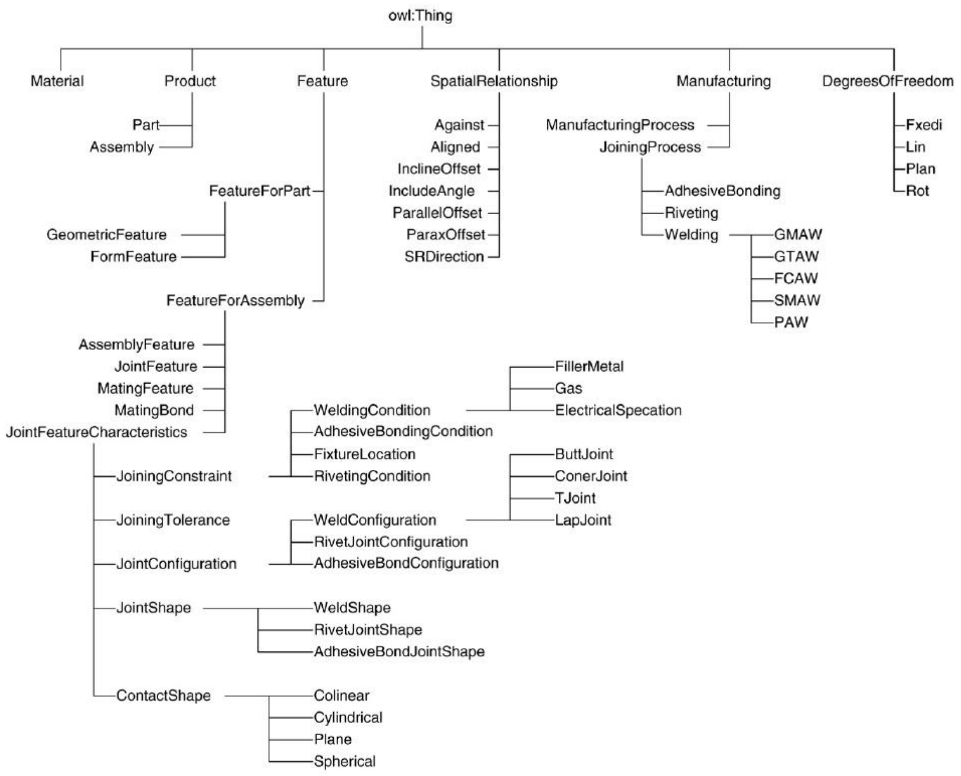


Figure 16. AsD ontology class hierarchy (Kim, Manley, Yang, 2006)

5. Manufacturing Assets/Services Discovery and Composition Methods for Interoperability of Virtual Factory

Digital factory aims at using digital technologies to promote the integration of product design processes, manufacturing processes, and general collaborative business processes across factories. An important aspect of this integration is to ensure interoperability between machines, products, processes, and services. A digital factory consists of a multi-layered integration of the information related to various activities along the factory and related resources. Actors can fall in different categories, being humans (i.e., final users or participants in the production process), information systems or industrial machines. These physical entities must have a faithful representation in the digital world, usually referred to as digital twins.

A digital twin (DT) exposes a set of services allowing to execute certain operations and produce data describing its activity. We can imagine these data stored in a factory data space together with other information, e.g., data available from the company and production history, worker suggestions and preferences. Such services are typically used to query or manipulate the state of the system, and associated data are leveraged for diagnostics and prognostics. The availability of DT services and data can have a huge impact on the design of manufacturing processes, by allowing automatic recovery and optimization, and even automatic composition of the intermediate steps for achieving a production goal.

Inspired by the research about automatic orchestration and composition of software artefacts, such as Web services, we argue that: (i) an important step towards the development of new automation techniques in smart manufacturing is the modelling of DT services and data as software artefacts; (ii) the principles and techniques for composition of artefacts in the digital world can be leveraged to improve automation in the physical one.

A crucial difference between traditional software artefacts used in composition techniques and DTs is that DTs may not share the same view of the world. Modern information systems and industrial machines may natively come out indeed with their digital twin. In other cases, especially when the approach is applied to already established factories and production processes, digital twins are obtained by wrapping actors that are already in place. In such scenarios, data management techniques (including integration and exchange) are a key ingredient for DTs interoperability.

Properties of DTs. DTs are commonly known as a key enabler for the digital transformation in manufacturing, however, in the literature, there is no common understanding concerning this term. Different definitions agree on features such as (i) connectivity, i.e., the ability to communicate with other entities and digital twins, (ii) autonomy, i.e., the possibility for the DT to live independently from other entities, (iii) homogeneity, i.e., the capability, strictly connected to the autonomy, that allows to use the same DT regardless of the specific production environment, (iv) easiness of customization, i.e., the possibility to modify the behaviour of a physical entity by using the functionalities exposed by its DT, and (v) traceability, i.e., the fact that a DT leaves traces of the activity of the corresponding physical entity.

Therefore, the concept of DT can be seen as an evolution of the one of (Web) service, and it constitutes a cornerstone for integration in digital factories In this chapter, we therefore review the discovery and composition methods for interoperability of digital factories, by presenting a compact overview of the state-of-the-art of the scientific and technical research in these fields.

5.1. Service discovery

In IoT environments, as those ones deployed in digital factories, devices are connected to the Internet to provide services. IoT devices can be anything, i.e., sensors, RFID tag, RFID reader, actuators, etc. The different IoT devices may use different wireless/wired technology to transmit data to cluster head, i.e., WLAN, ZigBee, NFC, Zwave, WiFi. These devices may be different in sizes, computational capability; hence heterogeneity will be the major challenge. Users/clients access a service depending on their requirements, hence selecting the appropriate service from a large number of available services is an essential design parameter.

As discussed in (Shinde and Olesen, 2018), the traditional service discovery methods are not applicable to IoT environment due to its different requirements, hence it is important to review different discovery mechanisms. Shinde and Olesen (2018) specifically reviews different approaches and highlights recent ones, e.g., Pal et al, (2012) using Latent Semantic Indexing, Cassar et al. (2014) using Service Transaction Matrix and Latent Dirichlet Allocation, Meditskos and Bassiliades (2010) using Term Frequency and Inverse Document Frequency and web context generation, and Segev and Sheng (2012) using group-based service request forward approach, which are the more promising also for digital factories.

Another interesting recent survey is provided by (Abdellatif et al., 2018), in which the focus is on the architecture. In directory based architecture (i.e. centralized architecture), a common entry point for users and service providers is introduced. Its main functionality is the registration and selection of services. This type of solutions often suffers from a lack of scalability, and fault tolerance. Researchers addressed this problem by distributing the workload on a number of interconnected cooperating directories (i.e. distributed architecture), which guarantee a better performance. A wide range of solutions make use of P2P connection, between these distributed directories, while other solutions maintain a master-slave relation, forming a hierarchy of directories (i.e. hierarchical architecture), governed by a root element usually referred to as global directory.

In digital factories, hierarchical architectures or distributed architectures seem the most promising ones, especially taking into account their natural matching with Digital Twins.

5.2. Service composition.

As time passed, Web Service technologies matured and organizations started to embrace services to outsource common parts of their application functions. Moreover, Web Service composition has become the most promising way to support business-to-business application integration. Web Service composition refers to combining outsourced Web Services to offer new, value-added services. Composition differs from traditional system integration, i.e., via Enterprise Application Integration technologies – in which applications tightly engage each other in a white/grey box fashion – while Web Services promote integration in a black-box fashion. Several initiatives have been conducted to provide platforms, frameworks and languages to enable loose- coupled integration of heterogeneous systems, mostly for SOAP- based Web Services. This encompasses a wide set of standards, such as WSDL (syntactic service description), UDDI, OWL-S (semantic service specification), and BPEL for workflow-based representation of service compositions where bindings between services are known beforehand.

However, stakeholders disagreed in materializing these solutions, mainly due to a lack of widely accepted usage standards (Daniel et al., 2008; Bozkurt et al. 2013; Rauf et al., 2008). Particularly, organizations often develop and/or describe Web Services using different interface specification practices and concept models. From the specification practices point of view, unless appropriately specified by providers, service meta-data can be counterproductive and obscure the purpose of a service, thus hindering its adoption. These phenomena are known as anti-patterns, or indicators of poor-quality service interfaces (Rodriguez et al., 2012) that entangle definition and analysis of Web Services and compositions. From the concept models point of view, different approaches model Web Service compositions using domain specific languages and new modelling notations. Such ad hoc modelling methods are hard to learn and use, and often have a very limited tool support available (Rauf et al., 2008).

Recent surveys (Lemos et al., 2015; Mohr, 2016) include composition methods, specifically focusing on SOAP-based methods.

In the last few years, RESTful (REpresentational State Transfer) services appeared as a lightweight and cost- effective alternative for SOAP-based services. Lightweight RESTful services are designed to ease consumption, composition and building of community-driven services (mashups). Garriga et al., (2016) analyses and compare existing proposals in the RESTful service composition area. This area is fairly new and remains somewhat unexplored, evidenced by the lower number of mature proposals compared to SOAP-based service composition, but the area is growing at a rapid pace and has interesting practical implications. They have defined two sets of features to guide the analysis, covering functional and non- functional characteristics found in both SOAP-based and RESTful composition approaches.

Interestingly, their conclusions are as it follows:

- RESTful composition approaches are fairly new but a very fast significant maturation is expected in the following years. The first approaches appeared in 2008 and their number is gradually increasing year by year. Meanwhile, RESTful service composition is experiencing a transition to a robust and holistic framework. In addition, recent efforts have demonstrated the potential of integrating both SOAP-based and RESTful Web Services in a proper new service ecosystem or "Internet of Services", in which machine-readable descriptions allow automatic discovery, composition and communication of collaborative services. The adoption of SOA principles also allows for decomposing complex and monolithic systems into ecosystems of simpler and well-defined services. The use of principles such as common interfaces and standard protocols gives a horizontal view of an enterprise system.

  Notably the current trend of technologies for Digital Twins is to expose their functionalities via RESTful interfaces, so basically all the evolution of RESTful composition can be leveraged in the digital factory domain.

- They found different perspectives in the "REST vs. SOAP" debate. However, according to different authors, the alleged debate seems meaningless, since RESTful and SOAP-based services have different objectives, are better suited for certain contexts, and can even be used in conjunction (Vinoski, 2007; Pautasso et al., 2008). Individually, RESTful services can be more scalable, reliable and visible, and better fitting for ad hoc integration at Internet-scale. However, composition approaches built upon RESTful services may lack some of these properties – e.g., scalability. SOAP-based services can be used for enterprise level application integration where QoS, reliability and message-level security are critical (Pautasso et al., 2008; Lanthaler and Gutl, 2010). Recent efforts in large-scale legacy system migration to services have demonstrated the suitability of SOAP-based technologies and standards (Rodriguez et al., 2013). In this sense, both perspectives could benefit each other: incorporating concepts from SOAP-based composition will endow the Web 2.0 platform with a powerful and highly usable integration paradigm; while REST architectural principles – and the Web platform as a whole – may enrich the enterprise community by integrating RESTful services with traditional back-end systems (Rosenberg et al., 2008).

- RESTful services still suffer from shortcomings on semantically describing, finding and composing services as well as the absence of a holistic framework covering the entire service lifecycle. The main reason for these issues is the lack of an agreed standard to materialize

RESTful services and compositions. None of the proposed approaches or their satellite languages has gained broad support so far (Lanthaler and Gutl, 2010). Consequently, it is expected that research efforts in the area will focus on addressing these issues, which are indeed critical for the success of RESTful services and compositions. Probably specific standards for specific domains will emerge, e.g., in the Digital Factory domain.

6. FIWARE Overview

6.1. Overview

All information about FIWARE is summarized from (FIWARE Developers, 2019), (FIWARE Academy, 2019) and (Fox, J., 2019).

FIWARE is an open source platform for building smart solutions gather data from many different sources (including but not limited to IoT) to build a "picture" of the real world and then process and analyse that information in order to implement the desired intelligent behaviour (which may imply changing the real world) (FIWARE Developers, 2019). There are five components, namely context processing, analysis and visualization at the top of Figure 17; core context management (context blocker) at the middle top of Figure 17; Internet of Things (IoT), robots and third-party systems at the bottom of Figure 17; data/API management, publication and monetization at the right of Figure 17; and development tools at the left of Figure 17.

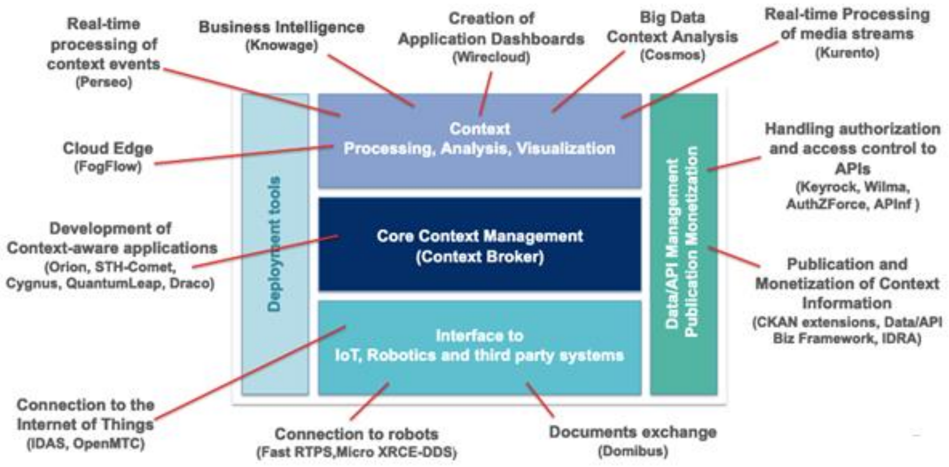

Real-time processing of context events (Perseo)
Business Intelligence (Knowage)
Creation of Application Dashboards (Wirecloud)
Big Data Context Analysis (Cosmos)
Real-time Processing of media streams (Kurento)
Cloud Edge (FogFlow)
Context Processing, Analysis, Visualization
Handling authorization and access control to APIs (Keyrock, Wilma, AuthZForce, APInf)
Development of Context-aware applications (Orion, STH-Comet, Cygnus, QuantumLeap, Draco)
Deployment tools
Core Context Management (Context Broker)
Data/API Management Publication Monetization
Publication and Monetization of Context Information (CKAN extensions, Data/API Biz Framework, IDRA)
Interface to IoT, Robotics and third party systems
Connection to the Internet of Things (IDAS, OpenMTC)
Connection to robots (Fast RTPS,Micro XRCE-DDS)
Documents exchange (Domibus)

Figure 17: FIWARE platform architecture overview (FIWARE Academy, 2019)

Core Context Management (Context Broker) allows you to model manage and gather context information at large scale enabling context-aware applications (FIWARE Academy, 2019).

- Internet of Things (IoT), robots and third-party systems, defines interfaces for capturing updates on context information and translating required actuations.
- Data/API management, publication and monetization, implementing the expected smart behaviour of applications and/or assisting end users in making smart decisions.
- Context processing, analysis and visualization of context information, bringing support to usage control and the opportunity to publish and monetize part of managed context data.
- Deployment tools support easing the deployment and configuration of FIWARE or third-party components and their integration with FIWARE Context Broker technology.

Different components map into FIWARE GEs (Fox, J., 2019), i.e. development of context-aware applications (Orion, STH-Comet, Cygnus, QuantumLeap, Draco); connection to the Internet of Things (IDAS, OpenMTC); real-time processing of context events (Perseo); handling authorization and access control to APIs (Keyrock, Wilma, AuthZForce, APInf); publication and monetization of context information (CKAN extensions, Data/API Biz Framework, IDRA); creation of application dashboards (Wirecloud); real-time processing of media streams (Kurento); business intelligence (Knowage); connection to robots (Fast RTPS,Micro XRCE-DDS); big data context analysis (Cosmos); cloud edge (FogFlow); documents exchange (Domibus).

There is a need to gather and manage context information that allows the manufacturing process to be dynamic. The processing of that information and informing external actors, enables the information to actuate and therefore alter or enrich the current context in a virtual factory platform for the FIRST project. FIWARE allows for a pick and mix approach. We are not forced to use these complementary FIWARE components but could use other third platform components to design a hybrid platform for FIRST.

The FIWARE context broker component is the core of the FIWARE platform. It enables the system to perform updates and access to the current state of context. The Context Broker in turn is surrounded by a suite of additional platform components, which may be supply context data from diverse sources such as a CRM system, social networks, mobile apps or IoT sensors for example, supporting processing, analysis and visualization of data or bringing support to data access control, publication or monetization.

In the context broker tier, the CRM information could be provided by Shuangchi Industry Co Ltd, social or selling trend information may collect by GK software and KM software, information of mobile apps and IT sensors can collect from manufacturers, retailers, and suppliers.

In the Internet of Things (IoT) tier, robots and third-party systems, IoT access will supported by SSPU, KM, GK. CRM systems or KM MES systems may be provided by Shuangchi and KM respectively.

In the context processing, analysis and visualization of context information tier, BU and RuG provides collaborative business process compliance analysis and verification. SAPIENZA provides manufacture service discovery and composition services for building a virtual factory. UniMore, and SAPIENZA provide digital twine services. RuG can provide energy consumption and simulations.

In the data/API management, publication and monetization tier, Unimore, GK, KM and SSPU could provide further data and API management for supporting all FIRST partners.

In general, FIWARE context broker, Internet of things, data/API management tiers could support the FIRST data level interoperability. Supporting FIRST services/assets and process level interoperability need to locate at FIWARE context process, analysis and visualization of context information tier.

6.2. Further reading

Further readings related to FIRST interoperability framework development are:

- *FIWARE Architecture*
  https://forge.fiware.org/plugins/mediawiki/wiki/fiware/index.php/FIWARE_Architecture#FIWARE_Architecture_Overview
- *Context Management Architecture*
  https://forge.fiware.org/plugins/mediawiki/wiki/fiware/index.php/Data/Context_Management_Architecture#Architecture_Overview
  - FIWARE.ArchitectureDescription.Data.BigData
  - FIWARE.ArchitectureDescription.Data.ContextBroker
  - FIWARE.ArchitectureDescription.Data.CEP
  - FIWARE.ArchitectureDescription.Data.StreamOriented
  - FIWARE.ArchitectureDescription.Data.CloudMessaging
  - FIWARE.ArchitectureDescription.Data.OpenData
  - FIWARE.ArchitectureDescription.Data.SocialSemanticEnricher

    - FIWARE.ArchitectureDescription.Data.SocialDataAggregator
    - FIWARE.ArchitectureDescription.Data.MetadataStore
    - FIWARE.ArchitectureDescription.Data.ElectronicDataExchange
- *Core context management: fundaments*
  https://fiware-tutorials.readthedocs.io/en/latest/getting-started/index.html

7. KMMES and Distributed Data Interoperability

KM Manufacturing Execution System (KMMES) is a workshop-oriented management information system for manufacturing process management issues at the manufacturing enterprise workshop site, which is located between upper planning, design management system ERP/PDM and Underlying industrial control DNC/SCADA. Through the plan arrangement, instruction delivery, process control, information release and data acquisition of the manufacturing site, we can meet the clients of real-time visual control of the site, reduce errors, improve efficiency, continuously improve the manufacturing process and build a transparent workshop for enterprises from planning, design to manufacturing.

1. Workshop resource modelling

   KMMES adopts the idea of object-oriented, to build a digital workshop resource model. It includes personnel modelling, equipment modelling, material modelling, production calendar modelling and man-hour modelling, etc.

2. Equipment operation management

   KMMES combines DNC data acquisition and digital manufacturing terminal data acquisition to support equipment monitoring and management.

3. Advanced planning scheduling

   KMMES offers low-volume, multi-variety, multi-process production dispatch planning modules that use reliable, high-performance optimized scheduling algorithms to help companies improve product delivery capacity while also handling complex production management issues such as emergency insertion, etc.

4. Job execution management

   KMMES collects and generates process data in real time, and feeds them back to ERP/WMS and other systems to realize closed-loop visual control of the production process.

5. Quality tracking management in production site

   Real-time, accurate and comprehensive delivery of quality information helps companies track the quality of production site in real time.

6. Material tracking management.

   KMMES can manage raw materials, intermediate products, and finished products and tooling materials in the manufacturing process, and form linear inventory management in combination with planned scheduling; provide touch screen, barcode and other methods to collect material data to complete tracking of material status of the enterprise. Establish a timely and effective material distribution mechanism to reduce the standby time during manufacturing.

7. Signboard monitoring.

   After processing the information of enterprise resource status and the processing tasks in the MES system, the on-site production process of workshop can be reappeared in real time from multiple angles and in all directions in the form of a graphical interface and a statistical list, according to the categories of virtual workshop, equipment monitoring and working condition monitoring. The production process, at the same time, the on-site production can be analyzed statistically and various report functions can be provided.

8. System integration application and function extension.

KMMES system supports data interaction and sharing with systems such as ERP, PDM, and CAPP, provides secondary development platform and an open system interface (API function set and functional components) for the Windows standard.

7.1. KMMES Supports Distributed Data Interoperability Solutions

KMMES is responsible for receiving the planning instruction information from the upstream ERP (Enterprise Resource Planning) system, i.e., when and how many products are completed; and the product data management product information to be manufactured of the PDM system (Product Data Management) is sent to the MES , that is, what the product is like, including Product function, size, material, etc.; CAPP system (Computer Aided Process Planning) sends the product manufacturing method information to MES , that is, how the products are processed, the processing steps and the required process equipment (tools, fixtures, measuring tools, attachments) . In addition, the MES has data interoperability relationship with the shop floor information management system DNC (Distribution Numerical Control), the data acquisition and monitoring control system SCADA (Supervisory Control And Data Acquisition), and the warehouse management system WMS (Warehouse Management System), e.g., for example ,The MES distributes the CNC machining instructions in the CAM to the DNC system according to the production plan , and the CNC program will be downloaded to the CNC machine tool by DNC ; after receiving the information, the MES will arrange the production, and the current status of the workshop resources needs to be obtained during the scheduling (whether the equipment is normal or not ,Whether the personnel are on duty, whether the inventory is sufficient, etc.), and transfer the processing parameters to the processing equipment and testing equipment; after the processing is completed, the warehouse is entered and enters the ERP inventory management. The upstream and downstream of the MES are shown in Figure 18 below.

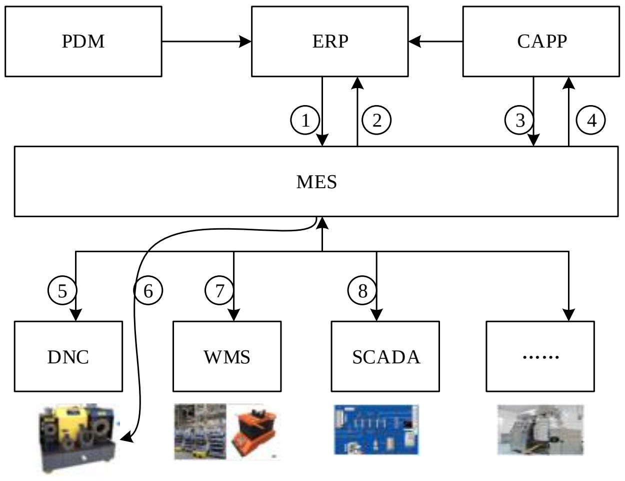


Figure 18. Logical relationship between MES and other information systems

7.1.1. Data Interoperability

Data Interoperability, also known as interoperability, refers to the ability of different computer systems, networks, operating systems and applications to work together and share information. There are different levels of interoperability, such as interoperability level syntax and semantic level of interoperability.

In terms of interoperability types, besides the ability of two or more computer systems to exchange information, semantic interoperability can automatically and efficiently interpret intentionally and accurately exchanged information to produce useful results defined by the end users of the two systems. In order to achieve semantic interoperability, both parties must refer to a common information exchange reference model. The content of the information exchange request is clearly defined: the content sent is the same as the content understood. The possibility of facilitating this result through user-driven fusion of different interpretations of the same information has become the subject of research prototypes (e.g. S3DB).

In the field of software engineering, interoperability is used to describe the ability of different programs to exchange data through a common set of exchange formats, read and write the same file format, and use the same protocol. Lack of interoperability may be the result of a lack of focus on standardization in the programming process. In fact, in the non-standards-based part of the computing world, interoperability is not taken for granted (Contesti, D.L., et al. 2007).

According to ISO / IEC 2382-01, the definition of interoperability in the basic terminology of information technology vocabulary is as follows: "In the case where the user is required to have little or no understanding of the unique characteristics of these functional units, the ability to establish communication mechanisms, execute programs or transmit data in each functional unit is required (SC36 Secretariat, 2003). This definition is somewhat ambiguous because the user of the program may be another program, and if the latter is part of an assembly that requires interoperability, it is likely that the program will need to be known.

Economic loss may occur if standard/specification data interoperability is not used. For example, studies show that the US capital facilities industry suffers from a loss of data usage costs of $15.8 billion annually due to insufficient interoperability. If competitors' products are not interoperable, the result is likely to be caused by monopoly or market failure (GCR, N., 2004). As a result, the user community or government has begun to take steps to encourage applications to adopt standards/specifications that support data interoperability, which has gradually become a trend for future application development. At present, at least 30 international agencies and countries have implemented e-government interoperability framework based Internet program called e-GIF, which has a similar NIEM program in the United States. The Standard Definition Organization (SDO) provides open public software specifications to facilitate interoperability; examples include the Oasis-Open organization and building SMART (formerly known as the International Alliance for Interoperability)(Transform, C.S., 2011). For the user community, the neutral third party is to create a standard for the interoperability of business processes, another example of neutral third parties is RFC documents from the Internet Engineering Task Force (IETF).

7.1.2. Interoperable Data Types Involved in KMMES

There are various data interoperability types in KMMES and ERP, PDM, CAPP and other systems. The related situations (relationship with other systems and interoperability types) can be explained in Table 3.

Table 3 data associated with the type of interoperability KMMES

serial number	**data interoperability related to KMMES**		**interoperable content**
	other systems	**interoperability type**	
1	ERP	ERP->KMMES	product orders, and product order changes routing, process code, work center, supplier material, inventory list, outbound order, etc.
		KMMES->ERP	completion information: task completion list, processing external agreement, processing quality QA, etc.
2	CAPP	CAPP->KMMES	NC program, process specification, technical documentation Process route, process

			Working hours, tooling tools ……
		KMMES->CAPP	process change information: part name/code, production plan number, quantity
3	DNC	DNC->KMMES	device status information: in the work piece name/code, processing start/end time, machine start / shutdown time processing information: spindle speed, feed rate alarm information: alarm start time, alarm number
		KMMES->DNC	machining instruction, NC program ......
4	machine tool	KMMES-> machine tool	NC program
5	WMS	KMMES->WMS	material information, quantity, current station
		WMS->KMMES	whether succeed
6	SCADA	KMMES->SCADA	process parameters of the device, such as temperature
		SCADA->KMMES	collected data for statistical analysis and display

7.1.3. Data Interoperability Mode between KMMES and Other Systems

In the manufacturing enterprise digital solution, KMMES employs three levels of data interoperability in terms of data availability. A brief description will be given below.

1. Sharing intermediate files

   In the early version of KMMES, data interoperability was supported by sharing intermediate files with tools such as CAPP. The technical path is to pre-agree the interoperable file format, storage location, and status change events (new, modified, and deleted). In application, the data provider according to the convention defined shared file create, maintain, or delete, and send state change events outward by the application after completion; after receiving messages, the data providers read the data according to the agreed shared intermediate file specification.

   This method often solves the data interoperability problem of the collaborative work group class. The interoperability type, data, scope, and permissions are all known, and the application scope is small and there are data security problems.

2. Shared database mode

   One of the data interoperability method still used in some of the early and currently used modules, that is, by sharing a database or some table files in the database. After the data provider completes creating, updating and maintaining data files, the component responsible for database access in the application sends a status event through the message; the data requester accesses the shared database or data table file according to the predefined permissions and scope.

   Compared with the shared file mode, the database method has better data security and reliability, can safely process the transmitted data in the storage process, and can support more complex distributed data interoperability applications.

3. Web service mode

   Web service is a platform-independent, low-coupling, self-contained, programmable-based web application development technology that uses open XML[1] (a subset of Standard General Markup language) standards to describe, publish, discover, coordinate, and supports the

[1] XML was created by the World Wide Web Consortium (W3C), and the XML SchemaXSD developed by the W3C defines a standard set of data types and gives a language to extend the set of data types. The main advantage of XML is that it is platform-independent and vendor-independent.

development of distributed data interoperability application system. Web Service technology supports different applications running on different machines to exchange data or integrate with each other without the aid of additional and specialized third-party software or hardware. That is, cross- language between applications implemented according to the Web Service specification the platform or internal protocol exchanges data.

In order to comply with the Web Service standard, the Web Service interoperability interface provided by KMMES uses XSD (XML Schemas Definition) as the basic data type , and uses WSDL (Web Services Description Language) to describe the Web Service and its functions, parameters and return values. Users use Web service UDDI (Universal Description, Discovery and Integration) mechanism to invoke the interface using SOAP protocol.

It is easy to deploy KMMES using Web Service to support distributed data interoperability, and can provide a common data interoperability mechanism for distributed collective enterprises or business process integration between multiple organizations.

7.1.4. KMMES Interoperability Framework

As mentioned above, there are three main modes of interoperability between KMMES and PDM , ERP , WMS , DNC , ACADA, etc.: rule-based file sharing, rule-based database sharing, Web service- based data sharing; and the interoperability framework divides the scope of sharing and physical segmentation through the workspace in Figure 19. Among them, the domain is divided into three levels of management, namely system management, security management, and audit management; the types of data access are: general business data, related data generated on the workflow, and data affected by data association. A brief description of the related concepts and key management activities is as follows.

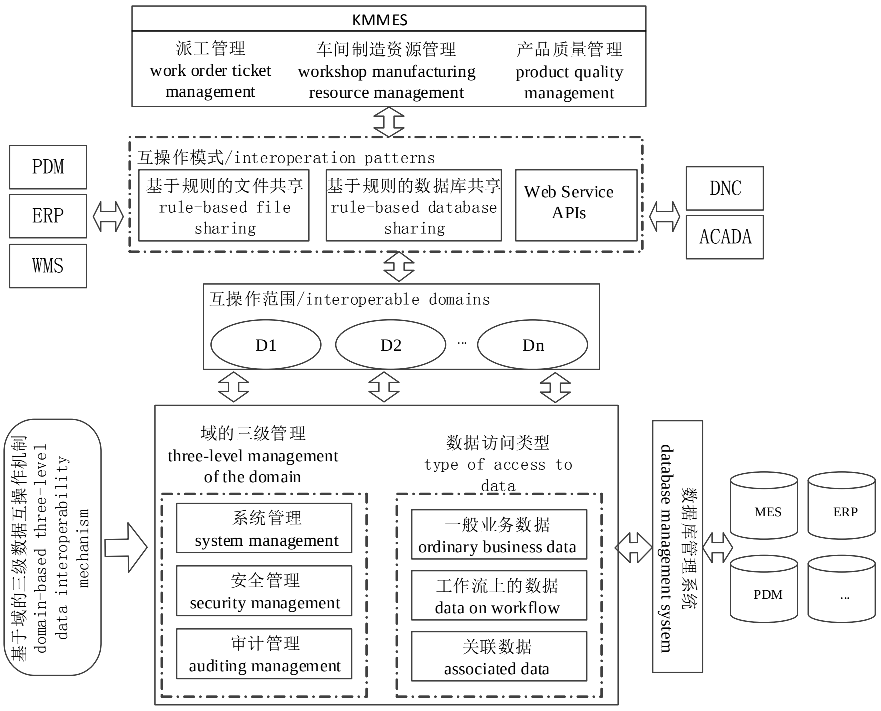


Figure 19. KMMES data interoperability framework model

- The concept of "domain" refers to the scope of an organization. The data in the domain, except for specific occasions or authorized, is generally available only to users in that domain. By default, there is a "primary domain" in the system. The domains created afterwards are all subdomains and the related roles of the primary domain, such as the system administrator, have administrative rights to the subdomain.

- Operating domain is done by the system administrator, including the creation and maintenance of each domain, and the initial password of the domain administrator.
- The primary domain and the subdomain all assign the super administrator's authority to the system administrator, security administrator, and security auditor (referred to as "three members"), and realize the separation of the three rights in each domain.
- The system sets up the primary domain administrator and multiple sub-domain administrators. The sub-domain administrators are responsible for independent management and control of the three members in each workshop; the primary domain administrator can only manage the sub-domain administrators of each domain, and cannot manage users in the sub-domain; the sub-domain administrator can only manage the domain in the region
- Data types that are isolated by domain include product and component structures, documents, related data objects, processes, projects, etc. Domain users create data in the domain, and the system records the domain ID. By default, data is separated by domain. Users in a domain can only operate the data of the domain they belong to in the normal authorization mode, but cannot see the data of other domains.
- In project management and process management, executors and roles can be assigned to roles and users in any domain. Regardless of the user of the domain, as the executor of the task, the task can be received in its task list and the object can be manipulated according to the permissions given by the task.
- The data between the domains is logically isolated. Through rules, cross-domain data sharing can be realized dynamically in projects and processes, supporting active sharing, automatic sharing of process and task triggers, and data sharing types based on data relationships.

7.2. KMMES Supports Data Interoperability Cases

7.2.1. Case Study 1 - a Beijing company

KMMES system is a workshop manufacturing execution system constructed and implemented by an enterprise in Beijing. It is a workshop-oriented production process management and real-time manufacturing information system. It implements KMMES and MRPII system (SAP) in China Airlines, open mesh Process Management System (MPM), and data transfer between the enterprise data centers, and the data flow of interoperability between systems is shown in the following Figure 20.

Interoperability mode: in this case interoperability is achieved by WEB Service.

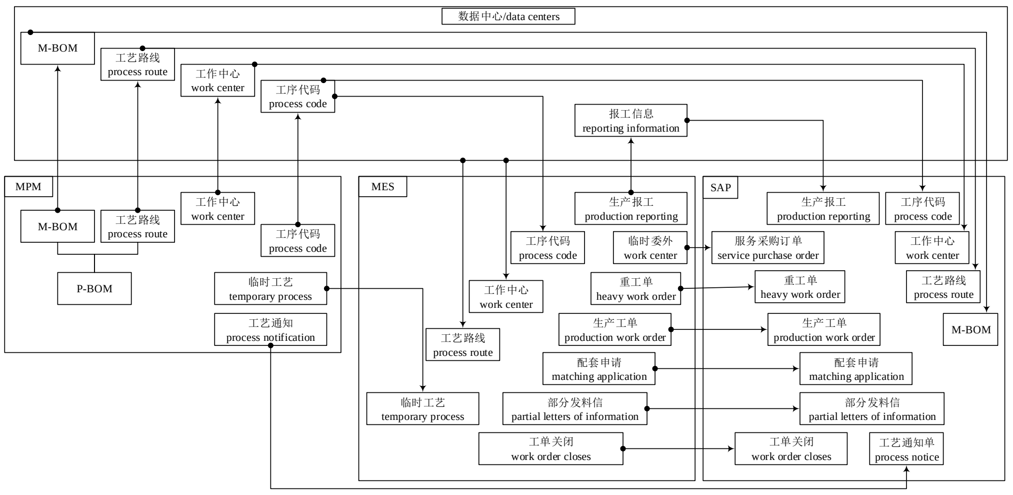
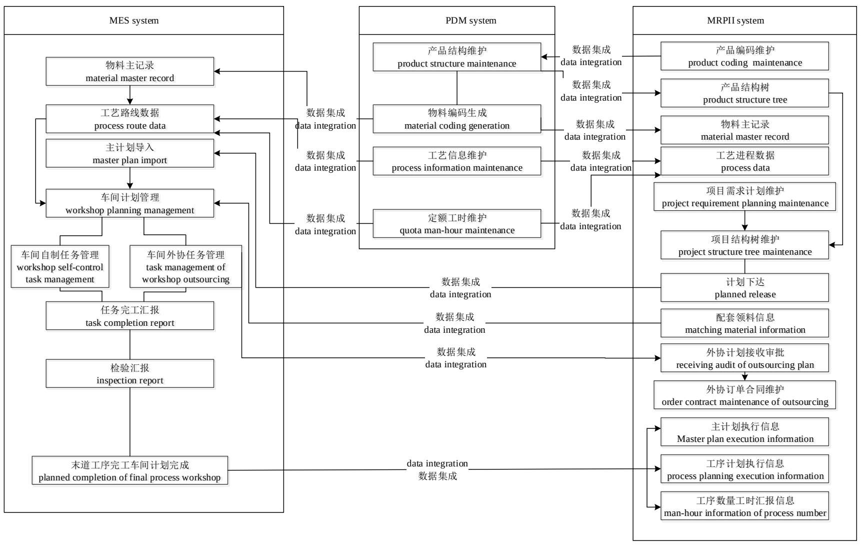

数据中心/data centers
M-BOM
工艺路线 process route
工作中心 work center
工序代码 process code
报工信息 reporting information
MPM
M-BOM
工艺路线 process route
工作中心 work center
工序代码 process code
P-BOM
临时工艺 temporary process
工艺通知 process notification
MES
生产报工 production reporting
工序代码 process code
临时委外 work center
工作中心 work center
重工单 heavy work order
生产工单 production work order
工艺路线 process route
配套申请 matching application
临时工艺 temporary process
部分发料信 partial letters of information
工单关闭 work order closes
SAP
生产报工 production reporting
工序代码 process code
服务采购订单 service purchase order
工作中心 work center
重工单 heavy work order
工艺路线 process route
生产工单 production work order
M-BOM
配套申请 matching application
部分发料信 partial letters of information
工单关闭 work order closes
工艺通知单 process notice

Figure 20. Business logic of Case1

According to the actual business work such as basic data preparation, plan release, task management, and execution record in the production management process, the interoperability business process of data in KMMES system, MRPII, and KMPDM system is as shown in the Figure 21:



MES system
物料主记录 material master record
工艺路线数据 process route data
主计划导入 master plan import
车间计划管理 workshop planning management
车间自制任务管理 workshop self-control task management
车间外协任务管理 task management of workshop outsourcing
任务完工汇报 task completion report
检验汇报 inspection report
末道工序完工车间计划完成 planned completion of final process workshop
PDM system
产品结构维护 product structure maintenance
物料编码生成 material coding generation
工艺信息维护 process information maintenance
定额工时维护 quota man-hour maintenance
MRPII system
产品编码维护 product coding maintenance
产品结构树 product structure tree
物料主记录 material master record
工艺进程数据 process data
项目需求计划维护 project requirement planning maintenance
项目结构树维护 project structure tree maintenance
计划下达 planned release
配套领料信息 matching material information
外协计划接收审批 receiving audit of outsourcing plan
外协订单合同维护 order contract maintenance of outsourcing
主计划执行信息 Master plan execution information
工序计划执行信息 process planning execution information
工序数量工时汇报信息 man-hour information of process number
数据集成 data integration

Figure 21. Data interoperability business process of case.

7.2.2. Case 2: A Research Institute of Aviation Technology

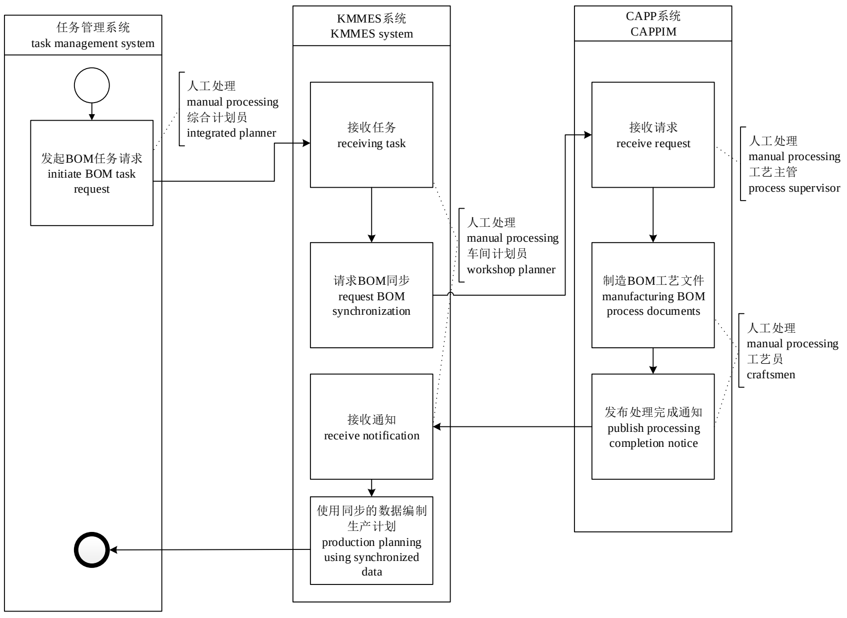
The interoperability scheme between the MES system and the KMPPIM (KMCAPP Integration Management) system in Figure 22 is as follows:

Interoperability mode: adopt Web Service + shared database + FTP mode;

Process online browsing uses the direct call KMPPIM system to provide OCX browser controls for browsing;

Data Interaction Step

- Baseline BOM (manufacturing BOM with process information) data interoperability



任务管理系统 task management system
发起BOM任务请求 initiate BOM task request
人工处理 manual processing 综合计划员 integrated planner
KMMES系统 KMMES system
接收任务 receiving task
请求BOM同步 request BOM synchronization
人工处理 manual processing 车间计划员 workshop planner
接收通知 receive notification
使用同步的数据编制生产计划 production planning using synchronized data
CAPP系统 CAPPIM
接收请求 receive request
人工处理 manual processing 工艺主管 process supervisor
制造BOM工艺文件 manufacturing BOM process documents
人工处理 manual processing 工艺员 craftsmen
发布处理完成通知 publish processing completion notice

Figure 22. Process of synchronizing BOM data between KMMES and KMCAPP.

1. MES system calls the interface provided by KMPPIM system to initiate a request to KMPPIM system; the request information includes graph number, version, name, stage mark, the design baseline number;
2. The KMPPIM system compiles the process of requesting figure number and all figure numbers of its subordinates. After the preparation, the complete material information, BOM structure relationship, process information (process number, process version) and process supporting information corresponding to the design baseline are stored in the shared database ,the two-dimensional and three-dimensional process files are stored in the ftp server, and the interface provided by the MES system is called to tell the MES system that the baseline information of the request synchronization has been compiled and put into the shared database.
3. After receiving the completed information, the MES system obtains the manufacturing BOM data from the shared database, including: material information used in the baseline, BOM structure relationship, process information (process number, process version), process matching information, and two-dimensional and three-dimensional process files information, which can be downloaded from FTP server.

- Batch process validation and delivery in Figure 23

1. The MES system calls the interface provided by the KMPPIM system to issue batch process confirmation request information to the KMPPIM system, the request information includes: graph number, version, number, batch number, process file number (before confirmation), and the process file version (before confirmation).

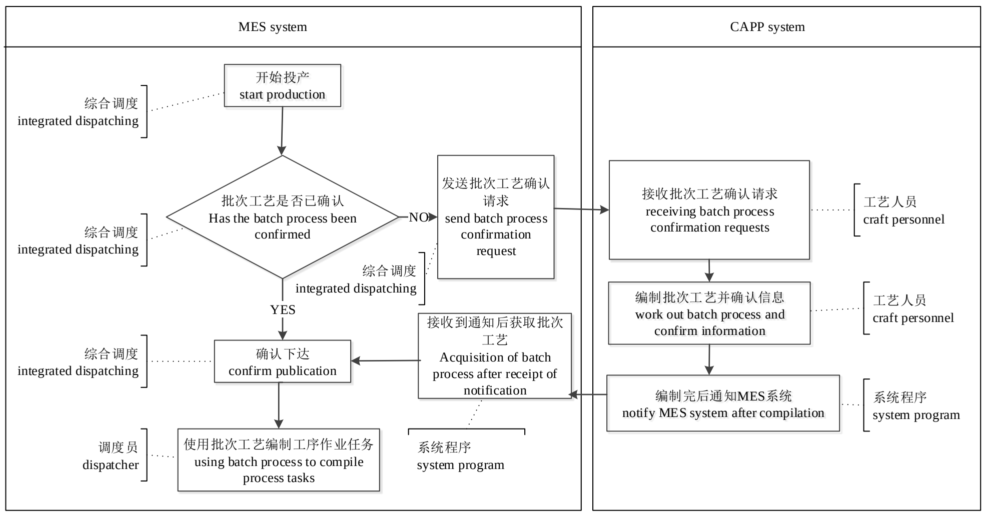

MES system
CAPP system
开始投产 start production
综合调度 integrated dispatching
批次工艺是否已确认 Has the batch process been confirmed
NO
YES
发送批次工艺确认请求 send batch process confirmation request
综合调度 integrated dispatching
综合调度 integrated dispatching
确认下达 confirm publication
接收到通知后获取批次工艺 Acquisition of batch process after receipt of notification
调度员 dispatcher
使用批次工艺编制工序作业任务 using batch process to compile process tasks
系统程序 system program
接收批次工艺确认请求 receiving batch process confirmation requests
工艺人员 craft personnel
编制批次工艺并确认信息 work out batch process and confirm information
工艺人员 craft personnel
编制完后通知MES系统 notify MES system after compilation
系统程序 system program

Figure 23. Confirmation and delivery process of batch process.

2. After receiving the batch process confirmation request from the MES system, the KMPPIM system compiles the process information of the requested batch in the KMPPIM system, and after compiling, the interface provided by the MES system is called to inform the MES system that the batch process confirmation request has been completed and confirmed, and the confirmation information includes: the figure number, the version, the quantity, number of process spare parts, batch number, process file number (after confirmation), and process file version (after confirmation).
3. After receiving the confirmation information, the MES system obtains the confirmed process information from the intermediate table, and the obtained information includes: figure number, version, quantity, number of process spare parts, batch number, process file number (after confirmation), and process file version (After confirming).
4. Standard process files are transmitted through the interface that passes the process files.
5. The batch process also uses a standard process file transfer interface.

- Process change in Figure 24

1. When a process change occurs in the KMPPIM system, KMPPIM system calls the interface provided by the MES system to push the process change order information to the MES system, change order information includes: change order number, figure number, version, process file number (after the change), Process file version (after change) , process file number (before change) , process file version (before change) , process file batch number (before change);
2. When the relevant person in charge of the MES system receives the notification of the process change, he can directly view the change order information and browse the process file information of the online change order.
3. The MES system summarizes all the current impacts scope of the change on the MES system according to the contents of the change order, and according to the actual system conditions of the MES system (in the manufacturing order, the warehoused, the assembled), implementation in the MES system (how to implement the change, the responsible person will make a decision);
4. On-site reorganization: (the method of reconciliation needs to be confirmed)

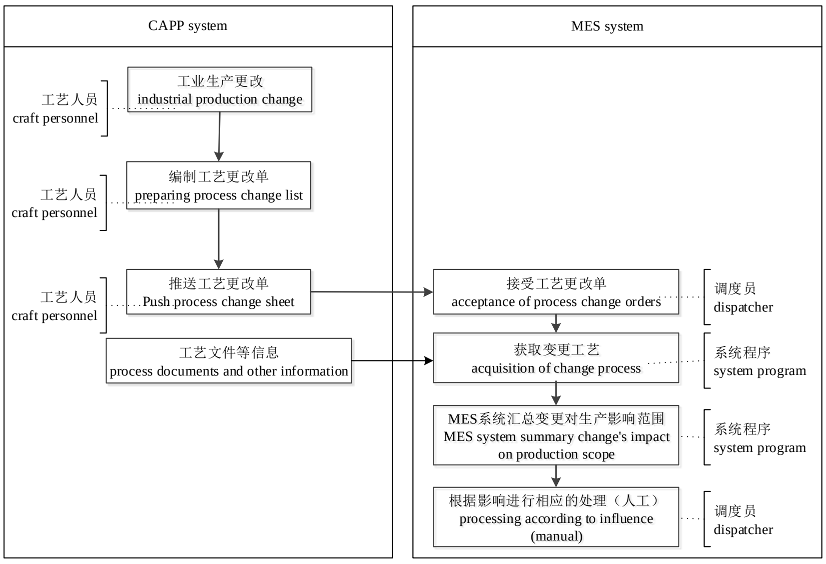

CAPP system
MES system
工艺人员 craft personnel
工业生产更改 industrial production change
工艺人员 craft personnel
编制工艺更改单 preparing process change list
工艺人员 craft personnel
推送工艺更改单 Push process change sheet
工艺文件等信息 process documents and other information
接受工艺更改单 acceptance of process change orders
调度员 dispatcher
获取变更工艺 acquisition of change process
系统程序 system program
MES系统汇总变更对生产影响范围 MES system summary change's impact on production scope
系统程序 system program
根据影响进行相应的处理（人工） processing according to influence (manual)
调度员 dispatcher

Figure 24. Process change process.

- Process online browsing

1. When the MES system displays the process file corresponding to the process on the spot, the positioning is performed according to the positioning parameter (page number), and the webpage plug-in provided by the KMPPIM system shows the content of the process file corresponding to the request parameter.

7.3. KMMES Data Interoperability Requirements for FIRST

Based on the ISA-95 standard, KMMES adheres to the philosophy of “information service production and manufacturing transparency”. By analysing the characteristics of discrete manufacturing, such as multi-variety and small batch, parallel production and development, independent process design and manufacturing, semi-automatic or manual equipment diversity, all aspects of production planning, manufacturing execution, quality management, material management and analysis and decision-making are managed in an all-round way. Establish professional digital manufacturing professional solutions oriented to discrete manufacturing enterprises, and realize closed-loop management of design process - manufacturing execution - feedback optimization.

The main problems that need to be solved are as follows:

- When the amount of data is large, it is necessary to further improve the real-time data acquisition and processing capability of the system, and has better data storage and access performance.
- Within the interoperability framework, when the state of the target data changes, a reliable message mechanism is needed to automatically notify the relevant data that the object is needed.
- In the wireless network environment, the integrity and reliability of data transmission should be ensured by using various protocols to process data transmission.
- System integration requires simple and convenient, involving less development.

8. Shoe Manufacturing

This section provides the current shoe manufacturing process of Shuangchi Industrial Co. Ltd. In this section, a case study is presented for showing currently shoe manufacturing processes in Shuangchi in Section 8.1. Analysis of the process limitations are also provided in Section 8.2.

8.1. Case Study: Shuangchi Shoe Manufacturing Processes

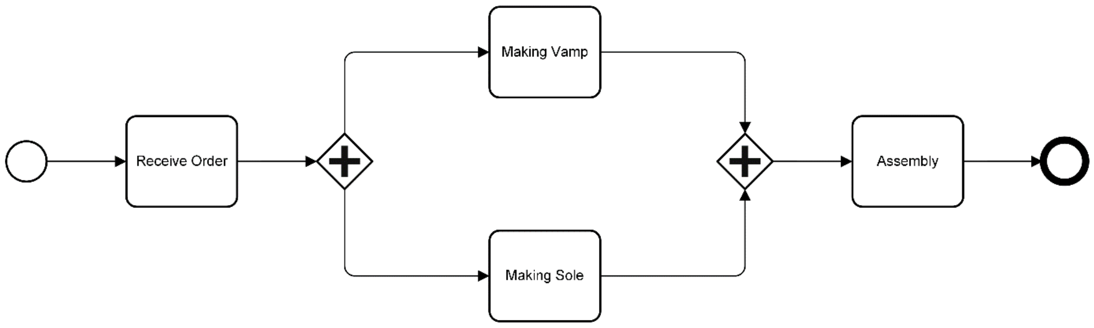
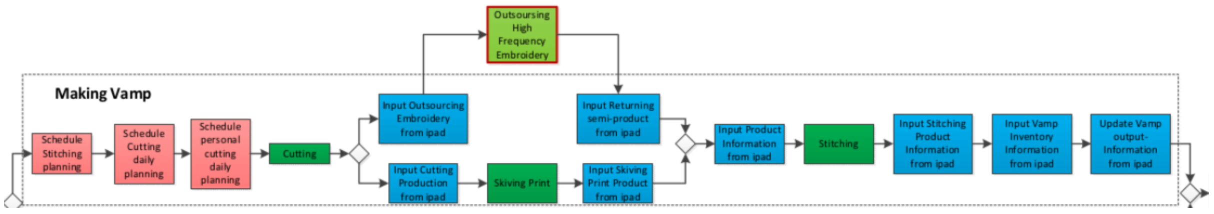


Figure 24. Shuangchi Co. Ltd Shoe Manufacturing Process

Figure 24 shows the shoe manufacturing process undertaken by Shuangchi Co. Ltd; from receiving orders through to shoe assembly and. The traditional process, as shown through colour coded tasks contains several manual operative tasks as well as manual activities. At a high-level view of the process, an order is received manually to which the schedule assembly is planned with the aid of computer support. From here, the overall process is split across two manufacturing processes; the making of the vamp and the making of the sole which are both split by an event-based gateway due the assembly process requiring both parts to be manufactured before being initiated in one instance.

8.1.1. Making Vamp

Figure 25 shows the sequence flow that manifests the vamp making process in more detail; time schedule planning for manual cutting is assisted by computer support and enables more effective resource and time planning for the execution of the cutting activity. Once cutting has been completed the process forks to outsource the high frequency embroidery and skive the cut leather, both of which rely on data input from an iPad in order to be executed.



Outsoursing High Frequency Embroidery
Making Vamp
Schedule Stitching planning
Schedule Cutting daily planning
Schedule personal cutting daily planning
Cutting
Input Outsourcing Embroidery from ipad
Input Returning semi-product from ipad
Input Cutting Production from ipad
Skiving Print
Input Skiving Print Product from ipad
Input Product Information from ipad
Stitching
Input Stitching Product Information from ipad
Input Vamp Inventory Information from ipad
Update Vamp output-Information from ipad

Figure 25: Low-level Vamp Manufacturing Sequence Flow

Shuangchi Industrial Co. Ltd does not facilitate the production of high frequency embroidery, meaning that this must be outsourced to a manufacturing company that specialises in the production of such embroidery and is enabled by data input from the iPad. The external is then completed by the acting external organisation and once completed, returning semi-product data is input from the iPad

Simultaneously, the activity of skiving the print is internally performed with cutting production data being inputted to from the iPad in order for the activity to be executed. Once this activity has been completed and the data has been inputted from the iPad, a parallel gateway joins the forked sequence flow to enable the manual completion of the stitching. However, stitching can only be performed once both the outsourcing of the embroidery and the skiving activities have been completed as both parts are required. Stitching is a manually performed activity which forms the vamp together and enables the sequence flow to progress towards the join prior to the shoe part assembly, in addition to input of key stitching and vamp inventory data from the iPad, as well as the vamp-output information

8.1.2. Making Sole

The production of the shoe sole is scheduled with the assistance of computer operation and enables the sequence flow to progress in Figure 27. Once assembly sole planning has been scheduled, the stock fitting progress is checked manually and thus the required stock fitting accessories are ordered manually following the progress check.

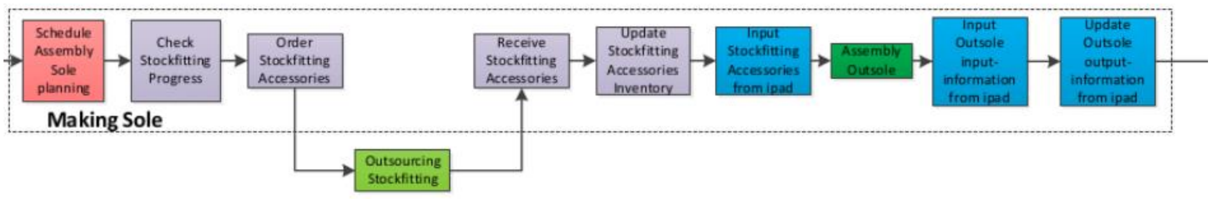

Schedule Assembly Sole planning
Check Stockfitting Progress
Order Stockfitting Accessories
Receive Stockfitting Accessories
Update Stockfitting Accessories Inventory
Input Stockfitting Accessories from ipad
Assembly Outsole
Input Outsole input-information from ipad
Update Outsole output-information from ipad
Making Sole
Outsourcing Stockfitting

Figure 26: Low-level Sole Manufacturing Sequence Flow

Shuangchi outsources just the stock fitting accessories, which are compiled and sent back to the company by a third party to be assembled to form the outsole, following the updating of the inventory and the accessory data input from the iPad. Assembly of the outsole in its entirety is performed manually and follows this the inputting of the outsole input information from the iPad and the update of the outsole output information from the iPad. Once these activities have occurred, the outsole in one instance is then prepared for the production of the shoe through its assembly with the vamp.

8.1.3. Assembly

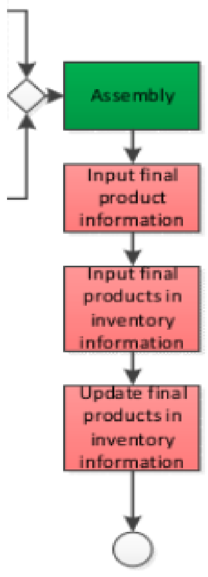
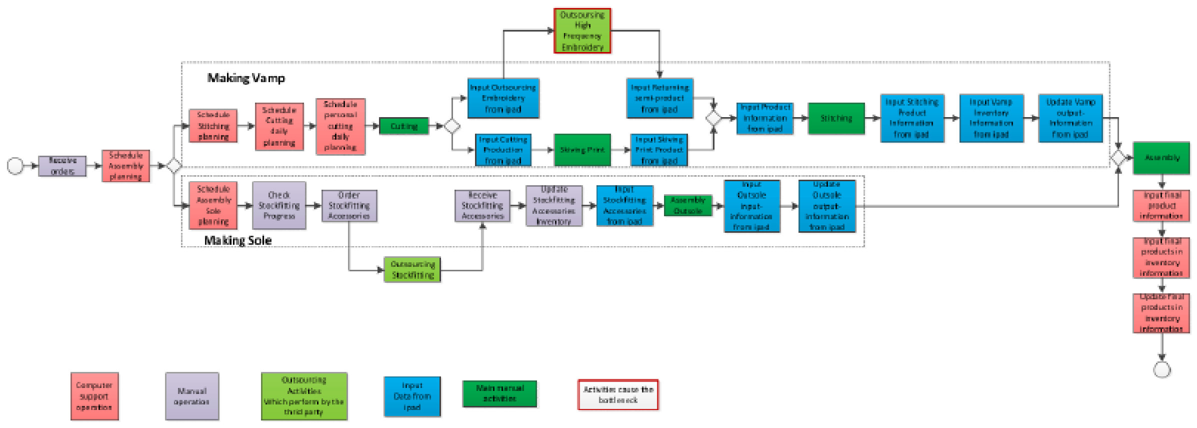

Assembly
Input final product information
Input final products in inventory information
Update final products in inventory information

Figure 27.Low-level Assembly Sequence Flow

Figure 27 shows the sequence flow manifesting the final activities performed in the production of the shoe; the completion of the Vamp and Sole joins the parallel gateway to allow for the manual assembly of the two parts to form the shoe. Once this activity has been completed, the final product information is inputted with support of computer function as well as the input of the final products in the inventory. The final activity performed in the shoe manufacturing sequence flow is the updating of the final products in the inventory and is done with the aid of computer support.

The overall show manufacturing process presented below in Figure 28:



Outsoursing High Frequency Embroidery
Making Vamp
Receive orders
Schedule Assembly planning
Schedule Stitching planning
Schedule Cutting daily planning
Schedule personal cutting daily planning
Cutting
Input Outsourcing Embroidery from ipad
Input Returning semi-product from ipad
Input Cutting Production from ipad
Skiving Print
Input Skiving Print Product from ipad
Input Product information from ipad
Stiching
Input Stitching Product Information from ipad
Input Vamp Inventory Information from ipad
Update Vamp output-Information from ipad
Assembly
Input final product information
Input final products in inventory information
Update Final products in inventory information
Schedule Assembly Sole planning
Check Stockfitting Progress
Order Stockfitting Accessories
Outsourcing Stockfitting
Receive Stockfitting Accessories
Update Stockfitting Accessories Inventory
Input Stockfitting Accessories from ipad
Assembly Outsole
Input Outsole input-information from ipad
Update Outsole output-information from ipad
Making Sole
Computer support operation
Manual operation
Outsourcing Activities Which perform by the third party
Input Data from ipad
Main manual activities
Activities cause the bottleneck

Figure 28. Overall shoe manufacturing process

8.2. Analysis of the current processes

The current shoe production sequence flow is absent of a logistic process; shown in Figure 29 where outsourcing occurs as a single activity outside of the product line, with embroidery and stock fitting being outsourced on an ad hoc basis. Stock fitting is traditionally completed outside of the main assembly line (Shoe Dog, 2015) and thus the outsourcing of the accessories, although limited by lack of a logistic process as shown in Figure 29, assembly is a main manual activity and as such manifests a potential bottleneck in the manufacturing process, primarily due to there being an increased scope for demand to be outweighed by production capacity as a result of a sole reliance on manual labour.

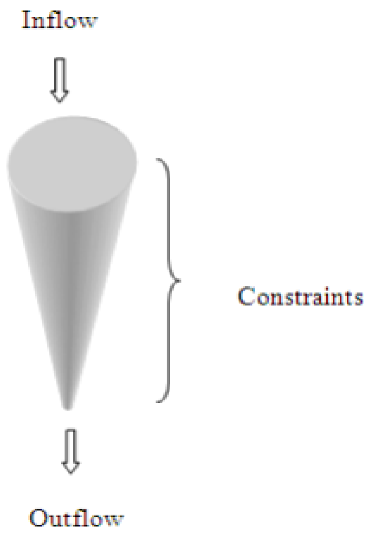


Figure 29. "Example of a Bottleneck" (Timilsina, 2012)

Timilsina (2012) looks at the different types of constraints faced in the manufacturing process and presents the concept of “people constraints”, a consideration of differentiation between people in terms of their motive to work, and how this can cause variances in individual work rate, in addition to unexpected vacancies or staff turnover and the issues that poses in terms of hiring and training as two examples. Accumulation has a higher potential to occur as a result of this and limit the productive capacity of the traditional shoe manufacturing process.

Such an identification can also be said for the remaining main manual activities in addition to the already identified bottleneck in the logistic process that occurs through the outsourcing of the high frequency embroidery as part of the vamp-making process as identified in Figure 25; cutting, skiving print, assembly outsole and stitching. All of which, as with assembly are vulnerable to people constraints. In addition to this, outsourcing is also required as part of the sole and vamp making processes and also provides a bottleneck in the overall process and the limitations caused by this are reviewed to elicit requirements for process improvement.

8.2.1. Supplier Process Limitations

A key issue with the provided model however, is that currently, there is no logistic process involved in the production of the shoes and this can be seen in the identification of the outsourcing of the high frequency embroidery as a single activity. The single activity shows that embroidery is currently ordered on an ad-hoc basis from a third-party supplier. The lack of a logistic process causes a bottleneck; embroidery is only sourced when it is needed and thus the supply chain is limited to a reactive state. Being in a state where embroidery is only supplied if and when it is needed can cause delays in the production of the shoe as per Figure 28; constraints posed on the production line by the inability of the third-party suppliers to pre-empt the organisations demand for embroidery can cause a restriction to the outflow of the activity.

8.2.2. Manufacturing Process Limitations

The current manufacturing process currently makes limited use of computer support in its operation, resulting in several activities where data is manually inputted from the iPad and such manual input increases scope for error due to human interaction. Additionally, manually inputting data from the iPad brings forth temporal repercussions to the sequence flow.

The source of this issue is there is currently no computer-support as a foundation for the whole process, besides the computer support currently made use in the schedule planning activities. The lack of computer system support means there is currently no internal database to house key data on products as they progress through the production line and thus manual input of data is required.

Furthermore, the current manufacturing process consists of several manual operations, including key activities that have a direct effect on the outcome of the shoes, such as cutting, skiving print, outsole assembly and stitching, in addition to those identified in as a manual operation by the key in Figure 29. With the introduction of smart factory concepts as previously discussed, relying on, manual operation throughout the manufacturing can cause additional bottlenecks where outflow of activities is less than their inflow. In addition to the greater risk of bottlenecks due to a heavy reliance on manual operation, a lack of automation also creates a higher risk of quality inconsistences due to human error.

8.2.3. End-Customer Limitations

The final stage of the production process sees the assembly of the shoe parts to form the shoe and following this, the activities that make up the input of the final product information to the inventory with support of a computer system. Currently however there is no logistic process for delivery of the product to the end customer due to the lack of integration logistic systems that enable to communication of the completion of an order and thus, it’s delivery.

9. Conclusions

In this deliverable we summarized existing work on existing interoperability framework for digital manufacturing platforms and related technologies so far. D1.3 deliverable is the first version of ‘‘Overview of existing interoperability of virtual factories’’ and provides the basis mainly for WP5, but also for WP2, WP3 and WP4. We have identified the research the issue how to dynamically switch between streaming data sources; provided potential approaches related to different manufacturing aspects of manufacturing assets/services classification for interoperability of virtual factory; as well as principles of manufacturing services discovery and composition methods for supporting interoperability.

Our academic and industrial partners have briefly summarized FIWARE platform and described how FIRST interoperability framework could be built based on FIWARE platform. KM software provides its detailed manufacturing solutions. A case study of Shuangchi shoe manufacturing processes and analyses the current process limitations are provided by Shuangchi Industry. Related interoperability requirements to FIRST framework are presented in this deliverable.

It is clear that the content of this deliverable can only be the starting point for extensive discussion on how manufacturing assets/service des are described, classified and discovered in the context of virtual factory to support on the fly business process verification, etc. This deliverable D1.3 provides the blueprint for D5.1. The initial version of D1.3 is going to be refined in a later release of this deliverable.

References

Otto, Boris, et al. "Reference architecture model Version 3.0." International Data Space Association https://www.internationaldataspaces.org/publications/reference-architecture-model-3-0, last accessed 2019/07/26 (2019).

Otto, Boris, et al. "Industrial Data Space: digital sovereignty over data" https://www.fraunhofer.de/content/dam/zv/en/fields-of-research/industrial-data-space/whitepaper-industrial-data-space-eng.pdf last accessed 2019/07/26 (2016)

Wajid, U. and Bhullar, G., 2019. Towards Interoperability Across Digital Manufacturing Platforms. In Enterprise Interoperability VIII (pp. 81-91). Springer, Cham.

FIWARE. (2018). Smart Industry - FIWARE. [online] Available at: https://www.fiware.org/community/smart-industry/, last accessed 26 July. 2019.

CEF Digital. (2019). Orion Context Broker. [online] Available at: https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/Orion+Context+Broker, last accessed 26 July 2019.

Min Chen, Shiwen Mao, and Yunhao Liu. (2014) “Big Data: A Survey”. In: Mob. Netw. Appl. 19.2 Apr. 2014, pp. 171–209. issn: 1383-469X. doi: 10.1007/s11036-013-0489-0. url: http://dx.doi.org/10.1007/s11036-013-0489-0.

Hashem, I.A.T., Yaqoob, I., Anuar, N.B., Mokhtar, S., Gani, A. and Khan, S.U., 2015. The rise of “big data” on cloud computing: Review and open research issues. Information systems, 47, pp.98-115.

Assunção, M.D., Calheiros, R.N., Bianchi, S., Netto, M.A. and Buyya, R., 2015. Big Data computing and clouds: Trends and future directions. Journal of Parallel and Distributed Computing, 79, pp.3-15.

Foster Provost and Tom Fawcett. (2013) “Data Science and its Relationship to Big Data and Data-Driven Decision Making”. In: Big Data 1.1, Mar. 2013, pp. 51–59. doi: 10.1089/big.2013.1508.

Karthik Kambatla et al. (2014) “Trends in big data analytics”. In: Journal of Parallel and Distributed Computing 74.7, 2014. Special Issue on Perspectives on Parallel and Distributed Processing, pp. 2561–2573. issn: 0743-7315. doi: https://doi.org/10.1016/j.jpdc.2014.01.003. url: http://www.sciencedirect.com/science/article/pii/S0743731514000057.

Apache Spark. (2019). url: https://spark.apache.org/ (visited on 06/22/2019).

E. Lazovik et al.(2017) “Runtime Modifications of Spark Data Processing Pipelines”. In: 2017 International Conference on Cloud and Autonomic Computing (ICCAC). Sept. 2017, pp. 34–45. doi: 10.1109/ICCAC.2017.11.

Apache Flink. (2019). url: https://flink.apache.org/ (visited on 06/22/2019).

Apache Storm. (2019). url: https://storm.apache.org/ (visited on 06/22/2019).

Apache Beam. (2019). url: https://beam.apache.org/ (visited on 06/22/2019).

Apache Hadoop MapReduce. (2019). url: https://hadoop.apache.org/ (visited on 06/22/2019).

Kubernetes Authors. Kubernetes. (2019) url: https://kubernetes.io/docs/home/ (visited on 06/22/2019).

CoreOS. Kubernetes Operators. (2019). url: https://coreos.com/operators/ (visited on 06/22/2019).

Apache Kafka. (2019). url: https://kafka.apache.org/ (visited on 06/22/2019).

Apache Storm. (2019). url: https://storm.apache.org/ (visited on 06/22/2019).

ISO 10303. The STEP Standard – ISO 10303. http://www.steptools.com/stds/step/

G. De Giacomo, D. Lembo, M. Lenzerini, R. Rosati, On reconciling data exchange, data integration, and peer data management, in: Proceedings of the Twenty-Sixth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 11-13, 2007, Beijing, China, 2007, pp. 133–142.

P. G. Kolaitis, Reflections on schema mappings, data exchange, and meta- data management, in: Proceedings of the 37th ACM SIGMOD-SIGACT- SIGAI Symposium on Principles of Database Systems, Houston, TX, USA, June 10-15, 2018, 2018, pp. 107–109.

M. J. Franklin, A. Y. Halevy, D. Maier, From databases to dataspaces: a new abstraction for information management, SIGMOD Record 34 (4) (2005) 27–33.

P. Cudre ́-Mauroux, M. T. Ozsu, Peer Data Management System, Springer US, Boston, MA, 2009, pp. 2055–2056.

V. Gadepally, P. Chen, J. Duggan, A. J. Elmore, B. Haynes, J. Kepner, S. Madden, T. Mattson, M. Stonebraker, The bigdawg polystore system and architecture, in: 2016 IEEE High Performance Extreme Computing Conference, HPEC 2016, Waltham, MA, USA, September 13-15, 2016, 2016, pp. 1–6.

J. Wang, T. Baker, M. Balazinska, D. Halperin, B. Haynes, B. Howe, D. Hutchison, S. Jain, R. Maas, P. Mehta, D. Moritz, B. Myers, J. Ortiz, D. Suciu, A. Whitaker, S. Xu, The myria big data management and analytics system and cloud services, in: CIDR 2017, 8th Biennial Conference on Innovative Data Systems Research, Chaminade, CA, USA, January 8- 11, 2017, Online Proceedings, 2017.

Lu, Qun, Chirpreet, Xu, & Ye. (2014). Ontology for manufacturing resources in a cloud environment. *International Journal of Manufacturing Research*, *9*(4), 448–469.

Panetto, H., Dassisti, M., & Tursi, A. (2012). ONTO-PDM: Product-driven ONTOlogy for Product Data Management interoperability within manufacturing process environment. *Advanced Engineering Informatics*, *26*(2), 334–348. https://doi.org/10.1016/j.aei.2011.12.002

Saha, S., Usman, Z., Jones, S., Kshirsagar, R., & Li, W. (2017). Towards a Formal Ontology to Support Interoperability Across Multiple Product Lifecycle Domains. *Proceedings - IEEE 11th International Conference on Semantic Computing, ICSC 2017*, 384–389. https://doi.org/10.1109/ICSC.2017.44

Szejka, A. L., Canciglieri, O., Panetto, H., Loures, E. R., & Aubry, A. (2017). Semantic interoperability for an integrated product development process: A systematic literature review. *International Journal of Production Research*, Vol. 55, pp. 6691–6709. https://doi.org/10.1080/00207543.2017.1346314

Nitishal Chungoora, Robert I. Young, George Gunendran, Claire Palmer, Zahid Usman, Najam A. Anjum, Anne-Francoise Cutting-Decelle, Jennifer A. Harding, Keith Case (2013). A model-driven ontology approach for manufacturing system interoperability and knowledge sharing. Computers in Industry, Vol. 64, pp. 392–401, Elsevier.

Anderson Luis Szejka, Osiris Canciglieri Júnior, Eduardo Rocha Loures, Hervé Panetto, and Alexis Aubry (2016). Proposal of a Model-Driven Ontology for Product Development Process Interoperability and Information Sharing. Proceedings of the International Federation for

Information Processing 2016 (IFIP). R. Harik et al. (Eds.): PLM 2016, IFIP AICT 492, pp. 158–168, 2016.

Liao, Y., Lezoche, M., Panetto, H., Boudjlida, N., & Rocha Loures, E. (2014). Formal semantic annotations for models interoperability in a PLM environment. In IFAC Proceedings Volumes (IFAC-PapersOnline) (Vol. 19, pp. 2382–2393). IFAC Secretariat.

Belkadi, F., N. Dremont, A. Notin, N. Troussier, and M. Messadia. (2012).A Meta-modelling Framework for Knowledge Consistencyin Collaborative Design. Annual Reviews in Control 36 (2): 346–358

Canciglieri Jr., O., and R. I. M. Young. (2010). Information Mapping across Injection Moulding Design and Manufacture Domains. International Journal of Production Research48 (15): 4437–4462.

Chen, Y. J. (2010). Knowledge integration and sharing for collaborative molding product design and process development. Computers in Industry, 61(7), 659-675.

Kim, K.-Y., D. G. Manley, and H. Yang. (2006). Ontology-based Assembly Design and Information Sharing for Collaborative Product Development. Computer-Aided Design 38 (12): 1233–1250.

Shinde, G. and Olesen, H., 2018. A Survey on Service Discovery Mechanism. In Intelligent Computing and Information and Communication (pp. 227-236). Springer, Singapore.

Paliwal, A. V., Shafiq, B., Vaidya, J., Xiong, H. and Adam N.(2012) Semantics-Based Automated Service Discovery. IEEE Trans. Serv. Comput., vol. 5, no. 2, pp. 260–275, Apr. 2012.

Cassar, G., Barnaghi, P. and Moessner, K. (2014) Probabilistic Matchmaking Methods for Automated Service Discovery. IEEE Trans. Serv. Comput., vol. 7, no. 4, pp. 654–666, Oct. 2014.

Meditskos, G. and Bassiliades, N. (2010) Structural and Role-Oriented Web Service Discovery with Taxonomies in OWL-S. IEEE Trans. Knowl. Data Eng., vol. 22, no. 2, pp. 278–290, Feb. 2010.

Segev, A. and Sheng, Q. Z. (2012) Bootstrapping Ontologies for Web Services. IEEE Trans. Serv. Comput., vol. 5, no. 1, pp. 33–44, Jan. 2012.

Abdellatif, S., Tibermacine, O., and Bachir, A. (2018) Service Discovery in the Internet of Things: A Survey. S. Chikhi et al. (Eds.): MISC 2018, LNNS 64, pp. 60–74, https://doi.org/10.1007/978-3-030-05481-6_5

Lemos, A. L., Daniel, F. and Benatallah, B. (2015) Web service composition: A survey of techniques and tools. ACM Comput. Surv. 48, 3, Article 33 (December 2015), 41 pages. DOI: http://dx.doi.org/10.1145/2831270

Mohr, F. (2016) Automated Software and Service Composition - A Survey and Evaluating Review. Springer Briefs,

Garriga, M., Mateos, C., Flores, A., Cechich, A., Zunino, A. (2016) RESTful service composition at a glance: A survey. Journal of Network and Computer Applications 60 (2016) 32–53

Daniel, F. and Pernici, B. (2008) Web service orchestration and choreography: enabling business processes on the web in e-business models, services and communications. In: Lee I, editor. IGI global; 2008. p. 250–73 chapter 12.

Bozkurt, M., Harman, M. and Hassoun, Y., (2013) Testing and verification in service- oriented architecture: a survey. Software Testing, Verification and Reliability, 23(4), pp.261-313.

Rauf, I., Iqbal, M., and Malik, Z. (2008) UML based modeling of web service composition – a survey. In: Proceedings of the international conference on software engineering research (SERA). IEEE; 2008. p. 301–7. Prague, Czech Republic.

Rodriguez, J.M., Crasso, M., Mateos, C., and Zunino, A. (2012) Best practices for describing, consuming, and discovering web services: a comprehensive toolset. Softw: Pract Exp 2012;43(6):613–39.

Vinoski, S. (2007) REST eye for the SOA guy. IEEE Internet Comput 2007;11(1):82–4.

Pautasso, C., Zimmermann, O., Leymann, F. (2008) RESTful web services vs. "big" web services: making the right architectural decision. In: Proceedings of the 17th international conference on World Wide Web (WWW). ACM Press; 2008. p. 805–14. New York, USA.

Lanthaler, M. and Gutl, C. (2010) Towards a RESTful service ecosystem. In: Proceedings of the 4th international conference on digital ecosystems and technologies (DEST). IEEE; 2010. p. 209–14. Dubai, UAE.

Rodriguez, J.M., Crasso, M., Mateos, C., Zunino, A., and Campo, M. (2013) Bottom-up and top-down COBOL system migration to web services: an experience report. IEEE Internet Computing 2013;17(2):44–51.

Rosenberg, F., Curbera, F., Duftler, M. and Khalaf, R. (2008) Composing RESTful services and collaborative workflows: a lightweight approach. IEEE Internet Computing 2008;12 (5):24–31.

FIWARE Developers (2019) [online] Available at: https://www.fiware.org/developers/ last accessed 31 July 2019.

FIWARE Academy (2019) [online] Available at: https://fiware-academy.readthedocs.io/en/latest/ last accessed 31 July 2019.

Fox, J. (2019) FIWARE Overview [online] Available at: https://www.slideshare.net/FI-WARE/fiware-wednesday-webinars-fiware-overview last accessed 31 July 2019.

Contesti, D.L., Andre, D., Henry, P.A., Goins, B.A. and Waxvik, E., (2007) Official (ISC) 2 guide to the SSCP CBK. CRC Press.

SC36 Secretariat (2003) Proposed Draft Technical Report for: ISO/IEC 2382, Information technology -- Learning, education, and training -- Management and delivery -- Specification and use of extensions and profiles: ISO/IEC 2382-01, ISO/IEC JTC1 SC36 N0646.

GCR, N., 2004. Cost analysis of inadequate interoperability in the US capital facilities industry. National Institute of Standards and Technology (NIST), pp.223-253.

Transform, C.S., 2011. E government interoperability: a comparative analysis of 30 countries. White paper by CS Transform, available at: www. cstransform. com.

Shoe Dog, T. (2015). Stock Fitting. [Blog] How Shoes are Made: The Sneaker Factory.

Timilsina, B., (2012). Removing bottleneck from a manufacturing unit: A case studies to BETKER OY, Ylivieska-84100, Finland.