

This is a pre print version of the following article:

Dynamic digital factories for agile supply chains: An architectural approach / Bicocchi, Nicola; Cabri, Giacomo; Mandreoli, Federica; Mecella, Massimo. - In: JOURNAL OF INDUSTRIAL INFORMATION INTEGRATION. - ISSN 2452-414X. - 15:(2019), pp. 110-121. [10.1016/j.jii.2019.02.001]

*Terms of use:*

The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

06/12/2024 01:58

(Article begins on next page)

# Dynamic digital factories for agile supply chains: An architectural approach

## Abstract

Digital factories comprise a multi-layered integration of various activities along the factories and product lifecycles. A central aspect of a digital factory is that of enabling the product lifecycle stakeholders to collaborate through the use of software solutions. The digital factory thus expands outside the company boundaries and offers the opportunity to collaborate on business processes affecting the whole supply chain.

This paper discusses an interoperability architecture for digital factories. To this end, it delves into the issue by analysing the key requirements for enabling a scalable factory architecture characterised by access to services, aggregation of data, and orchestration of production processes. Then, the paper revises the state-of-the-art w.r.t. these requirements and proposes an architectural framework conjugating features of both service-oriented and data-sharing architectures. The framework is exemplified through a case study.

*Keywords:* smart factory, digital factory, interoperability framework, process, service, data space

## 1. Introduction

Production processes are nowadays fragmented across different companies and organized in global multi-tier supply chains. This is the result of a first wave of globalization that, among the various factors, was enabled by the diffusion of Internet-based Information and Communication Technologies (ICTs) at the beginning of the years 2000. The recent wave of new technologies possibly leading to the fourth industrial revolution – the so called *Industry 4.0* – is further multiplying opportunities. Accessing global customers opens up great opportunities for firms, including small and medium enterprises (SMEs), but it requires the ability to adapt to different requirements and conditions, volatile demand patterns and fast changing technologies.

Supply chains are required to be more and more *agile*, where agility is defined as a combination of responsiveness and resilience. More specifically, *responsiveness* concerns the ability to adapt to changes in the demand, provide customers with personalized products (mass customization), quickly exploit temporary or permanent advantages and keep their competitive edge, while *resilience* concerns the ability to react to disruptions along the supply chain. The resulting agile supply chains will be able to successfully adapt to an evolving and uncertain business context in terms of both demand (customization, variability, unpredictability) and supply (new components, uncertainty in the supplies, bottlenecks and risks) taking into account not only the single organization but the entire value chain.

Our aim is to investigate methods and techniques for enhancing global multi-tier supply chains by addressing the methodological issues of how to apply digital technologies into existing supply chains and proposing a reference architecture.

*Digital factory* is a key paradigm to this end, as it aims at using digital technologies to promote the integration of product design processes, manufacturing processes, and general collaborative business processes across factories and enterprises [1, 2]. An important aspect of this integration is to ensure in-

teroperability between machines, products, processes, and services, as well as any descriptions of those. Accordingly, a digital factory consists of a multi-layered integration of the information related to various activities along the factory and related resources.

At the same time, leading institutions and firms in Europe, and specifically in Germany, have developed and published the Reference Architecture Model Industry 4.0 (RAMI 4.0) [3]. It describes the fundamental aspects of Industry 4.0 and aims to achieve a common understanding of what standards and use cases are required for Industry 4.0. Both the technological principles of digital factories and the RAMI 4.0 architectural principles are of particular importance for our purposes. However, there are still open challenges to be addressed in order to meet the requirements of agile supply chains.

In the following, we first introduce a case study scenario providing an exemplification of the main factors of an agile supply chain. Then, we overview the key contributions of this work.

### 1.1. The muffin factory application scenario

MyMuffin is a company operating within the EU producing muffins and willing to expand its business by allowing clients to buy muffins online. Clients can create their own muffins by picking pre-sets of ingredients and wait for delivery<sup>1</sup>. A client orders box(es) (each one containing 4 muffins) online, by choosing among different possible variants, such as: (a) chocolate chips vs. blueberry vs. apricot bits vs. carrot bits vs. nothing as additional ingredient; (b) butter cream vs. hazelnut cream

<sup>1</sup>MyMuffin is a fantasy company, but there are real successful examples of mass customization applied to food, cf. Mymuesli, a German company - <https://en.wikipedia.org/wiki/Mymuesli>. MyMuffin is an example of a small factory in which digital transformation can be applied in order to deeply modify production processes and business opportunities. Our work can be applied to such small factories as well as more complex ones, as in the automotive industry.

63 vs. icing sugar vs. nothing as topping; (c) yogurt vs. honey vs.119  
64 nothing in the dough. The client can also customize the colors120  
65 of the baking paper (wrapping the single muffin) as well as the  
66 colors of the box.

67 The muffin factory collects orders and organizes batches of121  
68 muffin doughs for production. As an example, if a client asks122  
69 for 3 boxes of carrot muffins with yogurt, icing sugar on top,123  
70 pink baking paper, and another client for 2 boxes of carrot124  
71 muffins with yogurt, nothing on top, yellow baking paper, the125  
72 same dough can be used for both orders. Clearly, this schedul-126  
73 ing service is based on the number of (and capacity of each)127  
74 dough mixers, the stream of received orders, etc. The factory128  
75 has a pool of dough mixers, of different capacity. The fact that129  
76 the number of different combinations is finite guarantees that130  
77 such a scheduling can be performed.

78 When an order is received, in parallel to the dough prepara-131  
79 tion, the baking paper should be set-up as well. In addition to132  
80 prepare a set of the requested paper baking cases, a QR-code133  
81 should be printed on each of them and used as a unique identi-134  
82 fier of the specific order. The identification of the single muffin135  
83 is crucial for customization. After the dough has been prepared,136  
84 the muffins are placed in the baking paper cases and sent to the137  
85 oven (connected to a QR code reader) for cooking. Muffins are138  
86 cooked in batches of about 1000 items and the length of this139  
87 step is equal for all of them.140

88 After the baking has been performed, the cart is operated141  
89 in order to route the different muffins to the right boxes, after142  
90 putting the right topping, and then to the proper delivery sta-143  
91 tion. Depending on the order, different delivery agents can be144  
92 used. Notably, agility is needed all along the process, e.g., the145  
93 baking step may overcook some muffins, which therefore are146  
94 not ready for the delivery and should be prepared again. This147  
95 imply a communication with the delivery agent in order to skip148  
96 the planned shipping and to set-up a new one and also a re-149  
97 scheduling of the mixers in order to re-introduce the preparation150  
98 of the damaged dough.

99 Figure 1 shows the process represented in Business Process151  
100 Model and Notation (BPMN), cf. <http://www.bpmn.org/>.152  
101 The reader not knowledgeable about BPMN can read a short153  
102 introduction about it in Section 2.1, where a detailed explana-154  
103 tion of the graphical notation adopted in the figure is also pro-155  
104 vided.156

## 105 1.2. Paper contribution and outline158

106 The main contribution of this paper is to provide a method-159  
107 ological and technological support to agile supply chains in the160  
108 Industry 4.0 context. To this end, it sets forth an architectural161  
109 framework that leverages RAMI 4.0 and addresses the method-162  
110 ological issue of making RAMI 4.0 capable of enabling agility163  
111 in supply chains.164

112 The proposed architectural framework enables interoperabil-165  
113 ity through a three-layered architecture where business pro-166  
114 cesses and goal descriptions trigger the discovery of the needed167  
115 services and data, and their composition in a dynamic, au-168  
116 tonomous and adaptive fashion.169

117 The rest of the paper is organized as follows: Section 2 pro-170  
118 vides an overview of the state of the art, Section 3 is the core171

section that presents the RAMI 4.0-based architectural frame-  
work, and finally Section 4 concludes the paper.

## 2. Background and related work

As pointed out in [4], pre-requisites for digital platforms to thrive in a manufacturing environment include the need for agreements on industrial communication interfaces and protocols, common data models and semantic interoperability. Currently, automated production plants, in fact, routinely employ thousands of devices from hundreds of vendors [5]. Furthermore, the growing importance of cooperation among organizations, encourages to dynamically establish inter-organisational interoperation.

In this situation, interoperability becomes a relevant challenge. Service Oriented Architectures (SOAs), Internet-of-Things (IoT) technologies, and open standards for device classification and discovery have been introduced to mitigate these issues [6, 7]. The most prominent examples of these trends are described in the following, whereas a detailed survey of the field is presented in [8].

Overall, despite the recent efforts aimed at the digitalisation of manufacturing, current approaches are still lacking in one or more of the following dimensions: (i) they still do not pursue a seamless integrated approach, which starting from processes arrives to data; (ii) they do not keep humans in-the-loop of product lifecycle management; (iii) they do not support in-process dynamic orchestration of services and data; (iv) they do not support alternative or personalised paths towards process goals.

### 2.1. Process management

Business Process Management (BPM) is a well-established discipline that deals with the identification, discovery, analysis, (re-)design, implementation, execution, monitoring, and evolution of business processes [9]. A business process is a collection of related events, activities, and decisions that involve a number of actors and resources that collectively lead to an outcome that is considered of value. Examples of business processes include order-to-cash, procure-to-pay, application-to-approval, claim-to-settlement, or fault-to-resolution.

To support business processes at an operational level, a BPM system (BPMS) can be used. As opposed to data- or function-centered information systems, a BPMS separates process logic from application code and, thus, provides an additional architectural layer. Typically, a BPMS provides generic services necessary for operational, software-enabled business process support, i.e., for process modeling, process execution, process monitoring, and user interaction (a.k.a. worklist management). When using a BPMS, software-enabled business processes are designed in a top-down manner, i.e., process logic is explicitly described in terms of a process model providing the schema for process execution. The BPMS is responsible for instantiating new process instances, for controlling their execution based on the process model, and for completing them. The progress of a process instance is typically monitored and traces of execution are stored in an event log and can be used for process mining

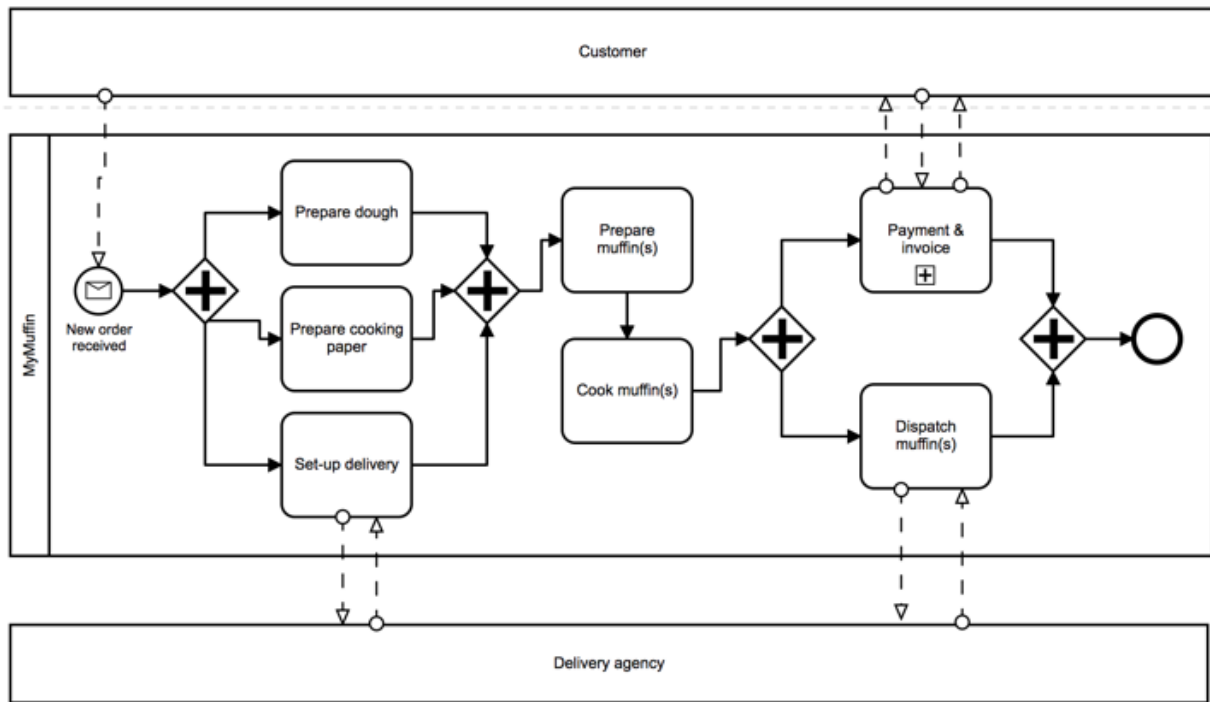


Figure 1: The process of MyMuffin. BPMN diagram, in which also public views of the delivery agency and the customer are shown as well (i.e., the whole supply chain).

[10], e.g., the discovery of a process model from the event log<sup>200</sup> or for checking the compliance of the log with a given process<sup>201</sup> model.

So far, the predominant paradigm to develop operational support<sup>202</sup> for business processes has been based on the *Model-Enact*<sup>203</sup> paradigm, where the business process has been depicted as a<sup>204</sup> (graphical) process model, which then could be executed by a<sup>205</sup> BPMS. This largely follows a top-down approach and is based<sup>206</sup> on the idea of a central orchestrator that controls the execution<sup>207</sup> of the business process, its data, and its resources.<sup>208</sup>

With the emergence of IoT, the existing Model-Enact<sup>209</sup> paradigm is challenged by the *Discover-Predict* paradigm; it<sup>210</sup> can be characterized as a bottom-up approach where data is<sup>211</sup> generated from devices sensing their environment and produc-<sup>212</sup> ing events. Sensor data must be then aggregated and interpreted<sup>213</sup> in order to detect activities that can be used as input for process<sup>214</sup> mining algorithms supporting decision-making [11].<sup>215</sup>

BPMN is the standard for business process modeling that<sup>216</sup> provides a graphical notation for specifying business processes<sup>217</sup> in a business process diagram (BPD), based on a flowcharting<sup>218</sup> technique. A diagram is constructed with a limited set of graph-<sup>219</sup> ical elements explained below, by using Figure 1 as an example.<sup>220</sup>

- Events, represented with circles, denote something that happens (compared with an activity, which is something<sup>222</sup> that is done). Icons within the circle denote the type of<sup>223</sup> event (e.g., an envelope representing a message, or a clock<sup>224</sup> representing time). In the example in Figure 1, the start<sup>225</sup> event of the process is when there is a *New order received*,<sup>226</sup>

and the process terminates when the flow reaches the bold-border circle.

- Activities, depicted as rounded rectangles, represent the single units of work. In our case study, they are *Prepare dough*, *Prepare cooking paper*, *Set-up delivery*, *Prepare muffin(s)*, *Cook muffin(s)*, *Dispatch muffin(s)* and *Payment & invoice*. Notably *Payment & invoice* is a sub-process, indicated by a plus sign against the bottom line of the rectangle, as it represents a compound activity, to be possibly detailed in its own diagram.
- Gateways, depicted with diamond shapes, determine forking and merging of paths. Exclusive gateways (showing an X inside the diamond) are used to create alternative flows in a process, as only one of the paths can be taken; parallel gateways (showing a + inside the diamond) are used to create parallel paths without evaluating any conditions. In the example, only parallel gateways are used, to mean that *Prepare dough*, *Prepare cooking paper* and *Set-up delivery* are all performed in parallel, then the flow is synchronized, and after some more activities performed sequentially, again *Dispatch muffin(s)* and *Payment & invoice* are performed in parallel.
- Connections are used to connect activities/events and gateways. (i) A sequence flow is represented with a solid line and arrowhead, and it shows in which order the activities are performed. As an example, *Prepare muffin(s)* is sequentially followed by *Cook muffin(s)*. (ii) A message

flow is represented with a dashed line, an open circle at the start, and an open arrowhead at the end. It tells us what messages flow across organizational boundaries (i.e., between pools – see further). A message flow can never be used to connect activities or events within the same pool. In the example, *Customer* sends a message to *MyMuffin* to start the process, messages are exchanged as well during the sub-process *Payment & invoice*. Analogously, messages are exchanged between *MyMuffin* and the *Delivery agency* during the activities *Set-up delivery* and *Dispatch muffin(s)*.

- Pools and lanes are used to represent participants in a process. In particular, each separate organization is represented as a pool (rectangle), as *Customer*, *MyMuffin* and *Delivery agency* in the example. A pool can contain one or more lanes, when the designer/modeler may want to organise and categorise activities according to a function or role within the same organization. A pool can be open (i.e., showing internal details, as *MyMuffin* in the example) when it is depicted as a large rectangle showing one or more lanes, or collapsed (i.e., hiding internal details, as *Customer* and *Delivery agency* in the example) when it is depicted as an empty rectangle stretching the width or height of the diagram. Notably, no specific functions/roles are depicted for *MyMuffin*, so no lanes are represented. When an organization is depicted as a collapsed pool, it is said to offer a *public view* of its processes, to mean that no internal details are exposed. In the example, *MyMuffin*, which is the subject of investigation, is completely modeled, whereas only the public views of *Customer* and *Delivery agency* are represented (i.e., their presence and the exchanged messages).

In the digital factory context, most of the works have been so far devoted to modeling issues, and specifically in the identification of suitable abstractions and modeling approaches and tools combining well-known standards with the specificities of digital factories, e.g., [12, 13]. Recently, the focus is shifting toward dynamicity during run-time, e.g., [14], in order to have automatic adaptation of production processes.

## 2.2. Service Oriented Architectures

A Service Oriented Architecture (SOA) is a valuable candidate for supporting integration among multiple conceptual layers and making distributed systems open and interoperable. Large enterprises promoted their use in manufacturing since late 90s [15]. A SOA offers the potential to provide the necessary system visibility and device interoperability in complex automation systems subject to frequent changes. A SOA can be considered as an architectural paradigm defining mechanisms to publish, find and compose services adopting loose coupling principles and open standards. It provides with technologies, methods and tools that can enhance interoperability by decoupling functionalities and their implementations. As a consequence, the transparency of the entire structure is increased, thus making the SOA paradigm particularly applicable in environments where reconfigurability is highly desirable.

Several recent EU research and innovation projects, such as SOCRADES [16], SODA [17], SIRENA [18], have demonstrated the feasibility of embedding web services at the device level and integrating these devices with a Manufacturing Execution System (MES) and Enterprise Resource Planning (ERP) system, at the upper levels of an enterprise architecture [19]. More in detail, SOA have been investigated for studying cross-organisational resource configuration [20], resource selection and utilisation [21] and Product Lifecycle Management (PLM, [22]). In [20], an agent-based software architecture for managing inter-organizational collaborations is proposed. A Colored Petri Net model specifying the role, which an organization fulfills in a collaborative process, is used to carry out the behavior of the agent representing the organization. In [21], it is proposed a solution for constructing a supply-chain information exchange platform. It adopts an heterogeneous data exchange engine and a data exchange agent to perform certain service functions such as end-to-end data exchange. In [22], a cloud-based framework capable of accommodating any kind of services and providing session control is proposed. More specifically, the framework enables services to collaborate with any combination of other services on the framework.

## 2.3. IoT technologies

The decentralised execution of self-organising and self-adaptive services has been recently discussed. In particular, the SAPERE project [23] conceptually models a service ecosystem as a virtual environment (e.g., a virtual factory). The interactions between services take place by applying a limited set of basic interaction laws, and typically take into account the spatial and contextual relationships between services.

IoT technology has been applied to the problem of service composition for improving both resource selection and utilisation [24]. More in detail, a configurable platform is proposed for the development of IoT-based applications, providing an information support base for both data integration and intelligent interaction in the product lifecycle, by combining ontologies and RESTful services. Based on an abstract information model, information encapsulating, composing, discomposing, transferring, tracing, and interacting in PLM can be carried out.

Though the composition of resource services is important, cross-organisation is seldom considered in such an environment. How a cross-organisational resource configuration impacts performance is discussed in [25].

Quality of service (QoS)-aware service composition in cloud manufacturing (CMfg) systems has also been proposed. As an example, the system proposed in [26] allows a free combination of multiple functionally-equivalent elementary services into a synergistic elementary service group to perform each subtask collectively, thereby improving the overall QoS. To deal with the increasing computing complexity of the optimisation model, an algorithm, named matrix-coded genetic algorithm with collaboratively evolutionary populations, has been designed.

A similar approach is discussed in [27]. A genetic algorithm was used to achieve global optimisation with regard to service level agreements – SLAs. Moreover, service clustering was

338 used for reducing the search space of the problem, and asso-393  
339 ciation rules were used for a composite service based on their394  
340 histories to enhance service composition efficiency. 395

#### 341 2.4. Asset description, classification and discovery 396

342 Device integration makes data and functionalities of devices397  
343 available throughout the entire automation system in ways that398  
344 support association, integration, data exchange, and possibly399  
345 semantic descriptions. Currently, the most widespread and rel-400  
346 evant technologies include Electronic Device Description Lan-401  
347 guage (EDDL), Field Device Tool (FDT)/Device Type Manager402  
348 (DTM) and Field Device Integration (FDI). 403

349 With FDI, a technology has been developed that combines404  
350 the advantages of FDT with those of EDDL in a single, scal-405  
351 able solution. FDI considers the various tasks over the entire406  
352 lifecycle for both simple and the most complex devices, in-407  
353 cluding configuration, commissioning, diagnosis and calibra-408  
354 tion [28]. Globally leading control system and device manufac-409  
355 turers, such as ABB, Emerson, Endress+Hauser, Honeywell,410  
356 Invensys, Siemens and Yokogawa, along with the major associ-411  
357 ations FDT Group, Fieldbus Foundation, HART Communica-412  
358 tion Foundation, OPC Foundation, PROFIBUS PROFINET In-413  
359 ternational, are supporting the development of the FDI together.414

360 In most scenarios, taxonomies are usually adopted as com-415  
361 mon ground for semantic interoperability. Classifying products416  
362 and services with a common coding scheme facilitates com-417  
363 merce between buyers and sellers and is becoming mandatory418  
364 in the new era of electronic commerce. Large companies are419  
365 beginning to code purchases in order to analyse their spending.420  
366 Samples of taxonomy including the description and classifica-421  
367 tion of manufacturing assets and services are: eCI@ss, UN-422  
368 SPSC, and MSDL [29]. 423

369 Nonetheless, this approach to semantic interoperability can-424  
370 not be employed in the considered agile application scenarios.425  
371 Indeed, most coding systems today have been very expensive426  
372 to develop and do not rapidly adapt to context changes. The ef-427  
373 fort to implement and maintain these systems usually requires428  
374 extensive utilization of resources, over an extended period of429  
375 time. Additionally, maintenance is an on-going and expensive430  
376 process. Another problem is that company's suppliers not nec-431  
377 essarily and always do adhere to the coding schemes of their432  
378 customers, if any. 433

379 With the increasing number of assets, service discovery be-434  
380 comes an integral part of digital factories. Service discovery435  
381 provides a mechanism which allows automatic detection of ser-436  
382 vices offered by any component in the system. The objective of437  
383 a service discovery mechanism is to develop a highly dynamic438  
384 infrastructure where requestors would be able to seek particu-439  
385 lar services of interest, and service providers offering those ser-440  
386 vices would be able to announce and advertise their capabilities.441  
387 Furthermore, service discovery should minimize manual inter-442  
388 vention and allows the system to be self-healing by automatic443  
389 detection of services which have become unavailable. Once444  
390 services have been discovered, devices in the system could re-445  
391 motely control each other by adhering to some standard of com-446  
392 munication. Over the past years, many organizations and major447

software vendors have designed and developed a large num-  
ber of service discovery protocols such as SLP, Jini, UPnP and  
UDDI.

#### 2.5. Data sharing and interoperability

A global multi-tier supply chain necessarily requires the in-  
terconnection among different and often heterogeneous infor-  
mation systems. From the perspective of data management, the  
main issue is to effectively manage heterogeneity in a dynamic  
context while preserving the autonomy of the data sources. In-  
deed, the different information systems offer data, information  
and knowledge from sources distributed over different stake-  
holders. All these sources are independent, making thus a-priori  
agreements unlikely.

Given a collection of disparate and distributed data sources,  
the main objective is often to provide a unified view (i.e., *data  
integration*) or to enable the exchange of data among them (i.e.,  
*data exchange*) [30]. In this context, the main difficulty lies in  
the fact that there is no agreement on the adopted data manage-  
ment 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 [31].

During the last two decades, many aspects concerning data  
sharing and interoperability have been studied including data  
management abstraction and architectures. The most interest-  
ing proposals that can be exploited to devise an interoperability  
platform for digital factories supporting agile and global multi-  
tier supply chains are (i) dataspace, (ii) peer data management  
systems, and (iii) polystores.

A dataspace [32] is an abstraction for data integration allow-  
ing the coexistence of heterogeneous data sources by providing  
basic functionalities over all data sources, regardless of how  
integrated they are. The goal is to reduce the effort required  
to set up a data integration system by relying on existing map-  
ping generation techniques, and to improve the system in a *pay-  
as-you-go* fashion. Dataspace principles can be thus exploited  
to manage dynamic situations. Moreover, the interaction with  
end users is the distinctive element of some pay-as-you-go ap-  
proaches for dataspace systems.

A peer data management systems (PDMS) [33] is defined  
as a set of autonomous peers exposing data and the related  
schema and a set of schema mappings. A PDMS therefore is  
a distributed data integration system providing transparent ac-  
cess to heterogeneous databases without resorting to a central-  
ized schema. Instead of imposing a uniform query interface  
over a mediated schema, a PDMS let the peers define their own  
schemas and the consequent reformulation of queries through  
mappings relating schemas.

Polystore systems [34, 35] have been recently proposed as a  
flexible architectural solution to data sharing and interoperabil-  
ity pursuing the *one-size-does-not-fit-all* philosophy. They en-  
able query processing over heterogeneous stores while guaran-

teeing full source autonomy, just-in-time transparent data trans-  
formation and support to multiple query interfaces.

### 3. Enabling interoperability in digital factories

The approach undertaken in this work is based on RAMI 4.0  
(cf. Figure 2). RAMI is a three dimensional reference architec-  
tural framework in the manufacturing industry domain devel-  
oped in Germany by leveraging EU initiatives and guidelines<sup>2,3</sup>.

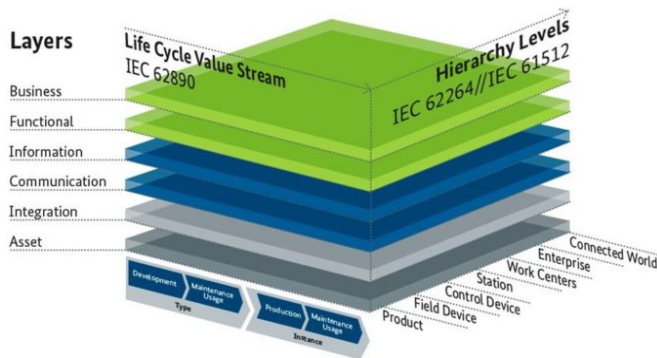


Figure 2: RAMI 4.0. A three dimensional reference architectural framework in the manufacturing industry domain.

We leverage RAMI 4.0 as the reference architectural frame-  
work describing how to apply digitalization technologies into  
existing supply chains to make them agile. According to RAMI  
4.0, data is the bridge towards digitalization and is described  
in the integration, communication and information layers. In  
global multi-tier supply chains, data characteristics are large-  
ness, distribution and heterogeneity. For instance, machines  
equipped with IoT sensors continuously produce data streams,  
OLTP data are available in DBMSs, OLAP data are available  
in data warehouses, digital manuals are stored in repositories  
and so on. To deal with these data features, data is organized  
in a dataspace of data sources that can exchange data through  
mappings. The dataspace adhere to the polystore architectural  
model supporting dynamic configurations (i.e. data sources go-  
ing in and out the system).

On top, at the functional level, different kinds of services are  
provided to get information and to perform actions on the man-  
ufacturing parts of system (e.g., producing and assembling ma-  
chines) as well as to enable interoperability with different ac-  
tors of the supply chain (e.g., order management, warehouse  
management). Open APIs are exposed by services in order to  
control, discover, and compose them in a dynamic way. Rich  
semantic descriptions of the services should be available in order  
to support both the discovery of the services and their exe-  
cution/invoke. This lays the foundations to achieve higher-  
level goals defined at the business level.

At the business level, in fact, business process specifica-  
tions must be able to capture not only orchestrated processes  
- which are bounded inside a single organization - but also  
choreographed processes which spans among different organi-  
zations, as a supply chain definition requires. Moreover, agility  
in the business processes can be achieved by shifting from the  
typical activity-centric process modeling to an artifact-centric  
modeling. This allows to model agile business processes with  
more emphasis to the goal to be achieved (i.e., the status to be  
reached) [36]. By defining several goals, with different degrees  
of completeness, the business process model is able to support  
a resilient and responsive environment, as the involved parties  
can tune their efforts to reach one of the goals, that is not nec-  
essarily the best one. Decisions on the goal to be achieved are  
driven by the available data [36].

One of the key issues to support agile supply-chains is to  
provide, manage and use the different services and data that  
are connected to the production processes. Manufacturing ma-  
chines typically provide data about their status and services.  
We face heterogeneous situations: from the one hand, ma-  
chines are from different vendors and, even if not proprietary,  
they are likely to adopt different standards and vocabulary; on  
the other hand, services can be provided at different levels of  
granularity, from very *fine grained* one (in terms of function-  
alities) to very *coarse grained*. As an example, the service of the  
oven may expose (simple, fine grained) operations for `start()`  
and `stop()`, whereas the scheduling service exposes a (com-  
plex, coarse grained) operation `schedule(listOfOrders):  
setOfMixerInstructions` which takes the list of received or-  
ders and return the set of instructions to be given for the dough  
to the different mixers. The role of the digital factory is to in-  
tegrate the different services and data and to combine them in  
order to make the whole process as efficient and competitive as  
possible in the achievement of the specific goals.

Another important issue to be faced is the fact that the pro-  
cess can cover a space wider than the single factory (it supports  
a supply chain): usually a factory gets the raw material from  
suppliers and provide products or semi-finished products to cus-  
tomers, through delivery agents, requiring the corresponding  
services and data to integrate to each other, or at least to be able  
to interact in a scalable and flexible way.

We envision a dynamic framework capable of assisting users  
through the discovery of service and data flows that best fit the  
expressed requirements and their evolution. The overall picture  
of the resulting RAMI 4.0-based architectural framework and  
the involved technological solutions are shown in Figure 3 . In  
the following the three layers are detailed.

#### 3.1. Process space layer - goal-oriented process specification

The top layer of the proposed architecture deals with the  
goals and the processes able to achieve such goals. In the My-  
Muffin example, some goals of the process are:

[G1] for each order, evade it within 36 hours (where evade  
means the muffins are packed and ready to be delivered);

[G2] for each order, the final delivery to the customer should  
be within 72 hours from the order.

<sup>2</sup>Cf. <https://www.plattform-i40.de/I40/Navigation/EN/InPractice/Online-Library/online-library.html>

<sup>3</sup>Cf. [https://ec.europa.eu/futurium/en/system/files/ged/a2-schweichart-reference\\_architectural\\_model\\_industrie\\_4\\_0\\_rami\\_4.0.pdf](https://ec.europa.eu/futurium/en/system/files/ged/a2-schweichart-reference_architectural_model_industrie_4_0_rami_4.0.pdf)

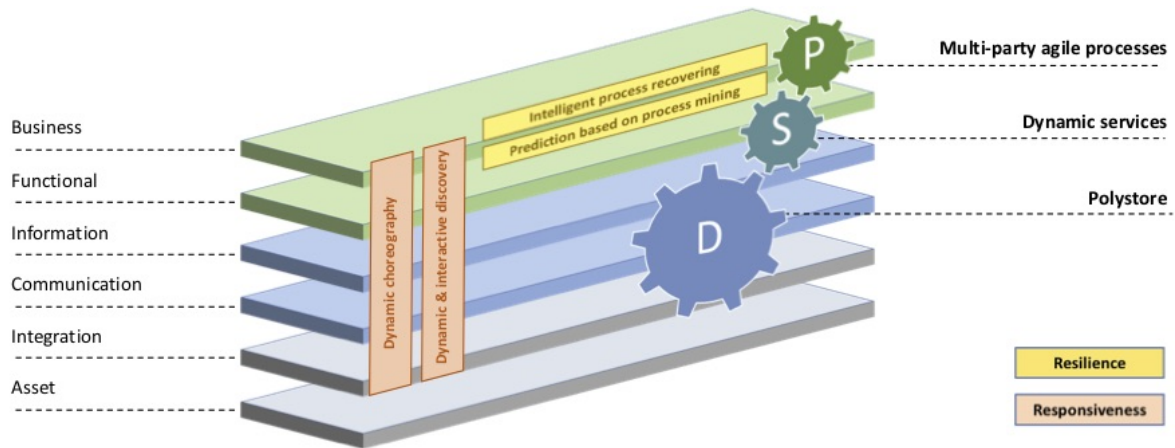


Figure 3: The enhanced RAMI 4.0. A dynamic framework capable of assisting users through the discovery of service and data flows that best fit the expressed requirements and their evolution.

The MyMuffin company adopts a process in which sub-goals might have been defined for specific parts (i.e., goals can turn be decomposed in sub-goals), e.g., in order to achieve G1 it should be

[G1.1] muffins should not be overcooked

Notably, MyMuffin would like to define, on the basis of such goals, specific KPIs – Key Performance Indicators, which qualify the QoS of the production process, e.g., the above 2 goals (i.e., G1 and G2) should be satisfied at least on 95% orders on weekly basis. Clearly, goals and KPIs are defined over many aspects, including the interactions with external companies being part of the process (e.g., the delivery agents having as goal to employ maximum 24 hours from pick-up to delivery, and to keep a KPI of 95% respected over the week).

As an example of agility, we can imagine that in a given day, some muffins get overcooked due to an error in the oven. This means that the goal [G1.1] is not achieved. In such a case, the digital platform will operate in order to re-arrange the process to achieve the goal. Through automated planning techniques, as the one adopted in SMARTPM [14], the process can be modified as shown in Figure 4. In particular, after the original activities *Prepare muffin(s)* and *Cook muffin(s)* (cf. Figure 1), new activities are introduced, in order to *Select alternative cooking service*, as a local bakery nearby MyMuffin that offers the availability of the oven; then, analogously to the original process, *Prepare dough* and *Prepare cooking paper* are performed, the muffins are moved and finally are received freshly cooked (cf. *Move muffin(s)* and *Receive freshly cooked muffin(s)*). Finally the process prosecutes as the original one. Notably, this is only one of the possible adaptations, the more complex it re-arranges the process; in the example, it is used if simpler solutions are not possible in the given situations. We will see later that other solutions at the underlying levels are possible, depending on the specific situation.

### 3.2. Service space layer - service discovery and composition

Starting from the goals and processes defined in the process layer, services must be dynamically composed to achieve

goal(s). In our example, we have different machines that can expose operations such as setting/increasing/decreasing the oven temperature, starting/stopping the dough mixer and providing related data by means of OpenAPIs. Rich semantic descriptions of the services should be available, in order to support both discovery and service execution. The descriptions should include some keywords that identify the context of the service (e.g., “food”, “cooking”), the equipment (e.g., “oven”, “mixer”), the performed operation (e.g., “turn-on”, “speedup”), and the parameters (e.g., “temperature”, “speed”).

With regard to the discovery phase, semantic descriptions are exploited to search for specific services without knowing their exact names and their syntax a priori. Semantic techniques can be exploited to find synonyms and keywords related to the words searched for in this phase. Searches can be performed either automatically by the process layer or by human operators which may be involved when needed (i.e., the adaptation techniques realized in the process layer fail, and a human intervention is needed in order to make the process progress).

Semantic descriptions can be used in the composition phase as well. Being the composition dynamic, the platform must not only find, but also use, the needed service or eventually provide support to human operators. To this purpose, the semantic description of the service parameters is needed in order to exploit the functionalities of the data layer to adapt the client service invocation to the server syntax (see next subsection). Some proposals and examples of semantic service descriptions have already been proposed [23].

The dynamism is useful to handle unexpected situations, often notified to a human operator. We report a couple of examples: in the former, an unexpected event causes an internal reorganization of the tasks; in the latter, an unexpected event deserves the interaction with an external actor.

The first example concerns oven performance. It may happen that the oven does not reach the required temperature due to different reasons (for instance, a cold winter day, bad isolation, broken door, and so on). The service provides `slowdown():delay`, which outputs the delay in percentage;



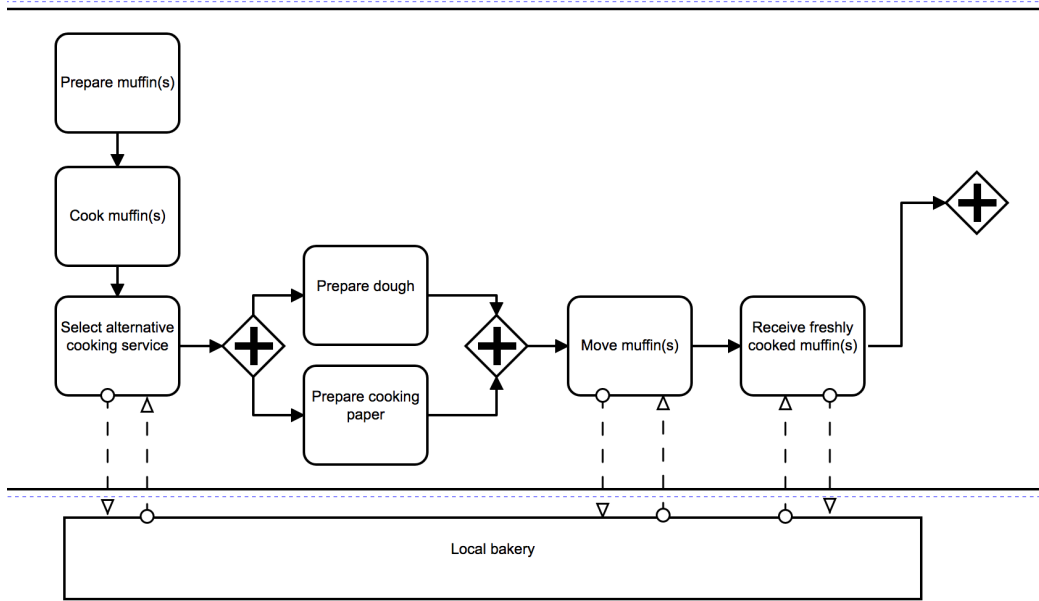


Figure 4: A fragment of the adapted process. After the original activities *Prepare muffin(s)* and *Cook muffin(s)* (cf. Figure 1), new activities are introduced, in order to *Select alternative cooking service*. Then, analogously to the original process, *Prepare dough* and *Prepare cooking paper* are performed, the muffins are moved and finally are received freshly cooked (cf. *Move muffin(s)* and *Receive freshly cooked muffin(s)*). Finally, the process prosecutes as the original one.

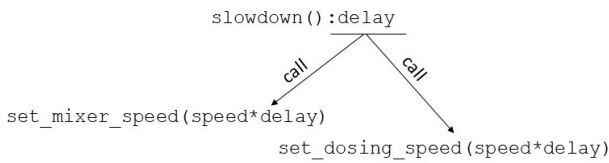


Figure 5: Service composition for adapting to oven performance.

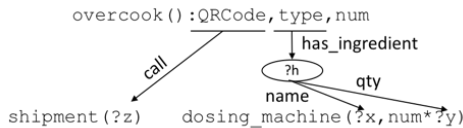


Figure 6: Service composition for the overcooked muffins.

for instance, if the oven was expected to reach the correct temperature in 30 minutes, but it actually needs 45 minutes, a delay of 33% is notified. The `slowdown()` is then composed with all the services available for reducing the speed of the machines; for instance, in Figure 5 `set_mixer_speed()` and `set_dosing_speed()` are invoked to reduce the speed of the dough mixer and of the dosing machine services.

The second example is more complex, even if related to a simple unexpected event: some muffins are overcooked, a case in which the shipping courier must be notified to modify the shipment and a new set of muffins must be produced starting from the list of needed ingredients. To this purpose, `overcook(): (QRCode, type, num)` is available and can be activated either by a monitoring facility or by human inter-

vention. This service outputs the type (cf. `type`) and number (cf. `num`) of the overcooked muffins and the corresponding order (identified by its QRCode) and can be composed with two discovered services: one interacting with the shipping courier (e.g., `shipment(URL)` with the courier Web service as input) and one activating the dosing machine (e.g., `dosing_machine(setOfIngredients, setOfQuantities)` with ingredients and quantities as input). The composition (see Figure 6) requires the connection of the output with the input. Essentially, the composition connects the discovered services by making explicit the relationships between the involved service parameters. `?x`, `?y`, `?z`, `?h` are variables and the corresponding values must be discovered in the data space as they represent the input to the two services for shipment and the dosing machine.

Clearly, the platform must also consider failure situations, such as oven out of work, refrigerator service not found, and so on. These issues require the frequent involvement of humans in the loop in order to deal with them in an effective way, or to revert to upper layers (as shown above in the case of complete process re-arrangement).

### 3.3. Data space layer - service-oriented mapping discovery and dynamic dataspace alignment

Data are managed and accessed in a data space. The data space must be able to deal with a huge volume of heterogeneous data by autonomous sources and support the different information access needs of the service level. In particular, a large variety of data types should be managed at the dataspace level. Data can be static such as data available in traditional

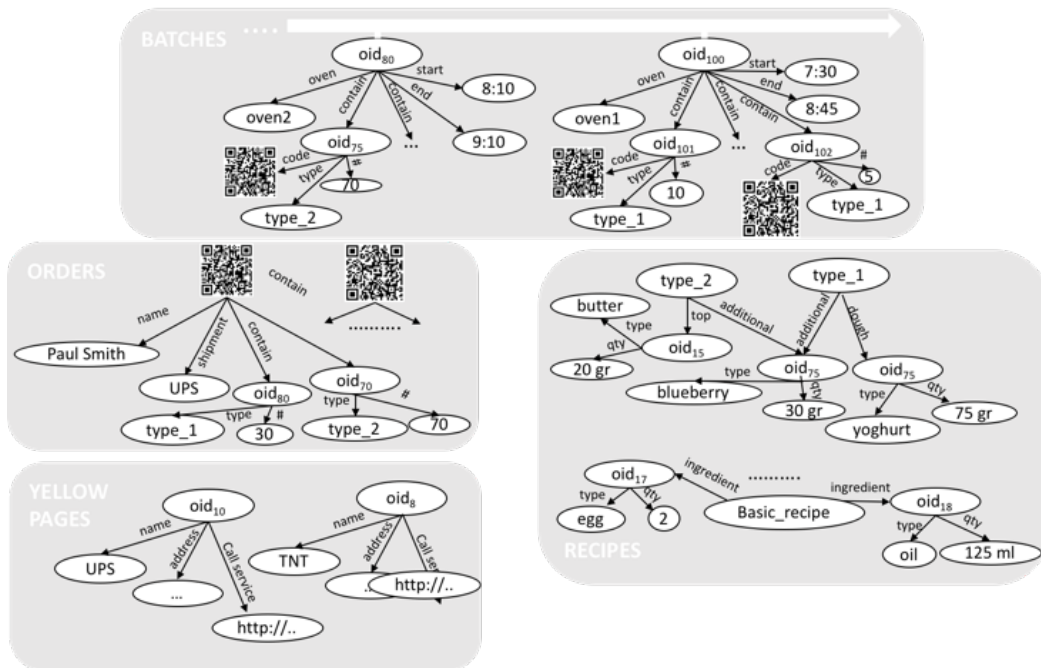


Figure 7: An excerpt of the MyMuffin data space.

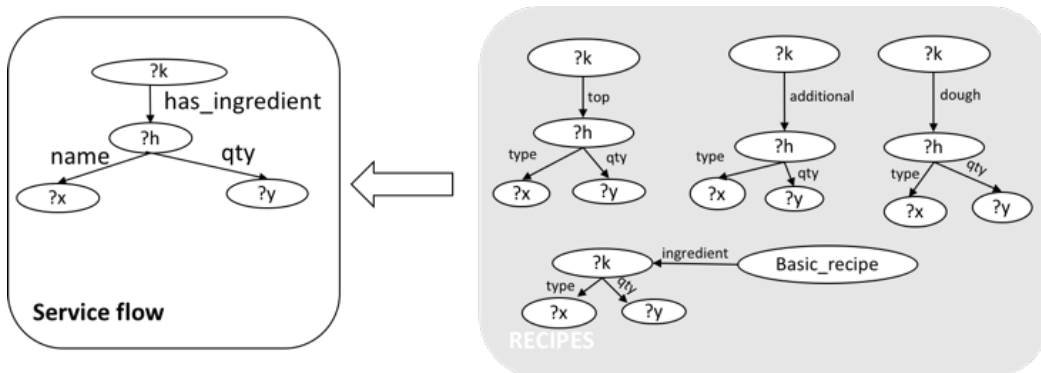


Figure 8: Mapping discovery process.

654 DBMSs but also highly dynamic like sensor data that are con-711  
655 tinuously generated. Moreover, the dataspace should accom-712  
656 modate data with different degrees of structure, from tabular to713  
657 fully textual data. Finally, the dataspace should cope with the714  
658 very diversified data access modalities sources offer, from low715  
659 level streaming access to high level data analytics. 716

660 To this extent, the data modelling abstraction we adopt to717  
661 represent the data space is fully decentralized, thereby bridg-718  
662 ing, on the one hand, existing dataspace models that usually719  
663 rely on a single mediated view [37] and, on the other hand, P2P-720  
664 approaches for data sharing [38]. The dataspace is therefore a721  
665 collection of heterogeneous data sources that can be involved in722  
666 the processes, both in-factory and out-factory. Those data are723  
667 either describing the manufactured products or the manufac-724  
668 turing processes and assets (materials, machines, enterprises,725  
669 value networks, and factory workers) [4]. Each data source has726  
670 its data access model that describes the kind of managed data,727  
671 e.g., streaming data vs. static data, and the supported operators.728  
672 As an example, sensed parameters such as temperature in the729  
673 oven, temperature in the packing station, levels of the different730  
674 ingredients, etc. are all streaming data needed in the dataspace731  
675 of MyMuffin that can be accessed only through simple window-732  
676 ing operators on the latest values. On the other hand, supplier733  
677 data can be recorded in a DBMS that offers a rich access model734  
678 both for On Line Transaction Process (OLTP) operations and735  
679 On Line Analytical Process (OLAP) operations. 736

680 Data representation relies on the graph modelling abstrac-737  
681 tion. This model is usually adopted to represent information in738  
682 rich contexts. It employs nodes and labelled edges to represent739  
683 real world entities, attribute values and relationships among en-740  
684 tities. Figure 7 shows a small portion of the MyMuffin data741  
685 space that can be used in case of overcooking. *Batches* is a742  
686 data stream that reports the cooking status over time; *Orders*743  
687 is the set of records storing the orders made by clients online744  
688 and the corresponding QR-codes; *Recipes* is a semi-structured745  
689 data set recording the recipes of the different kinds of muffins;746  
690 *Yellow pages* is a web-based data source about the couriers and747  
691 the related Web services. The *oids* in Figure 7, like *oid*<sub>101</sub>, are748  
692 object identifiers and are used to collect together data referring749  
693 to the same real-world entity. It is worth noting that graph data750  
694 can be serialized in a triple base where each triple has the form751  
695  $(s, p, o)$ , where  $s$  is the source,  $p$  is the property, and  $o$  is the752  
696 object. 753

697 The main issue that the interoperability platform must cope754  
698 with when dealing with data, is data heterogeneity. Indeed,755  
699 the various services gather data, information and knowledge756  
700 from sources distributed over different stakeholders and exter-757  
701 nal sources, e.g., the delivery agents and the Web. All these758  
702 sources are independent, and we argue that a-priori agreements759  
703 among the distributed sources on data representation and ter-760  
704 minology is unlikely in large digital supply chains over several761  
705 digital factories. 762

706 Data heterogeneity can concern different aspects: (1) differ-763  
707 ent data sources can represent the same domain using different764  
708 data structures; (2) different data sources can represent the same765  
709 real-world entity through different data values; (3) different766  
710 sources can provide conflicting data. The first issue is known767

as *schema heterogeneity* and is usually dealt with through the  
introduction of mappings. Mappings are declarative specifica-  
tions describing the relationship between a target data instance  
and possibly more than one source data instances. The sec-  
ond problem is called *entity resolution* (a.k.a. record linkage  
or duplicate detection) and consists in identifying (or linking  
or grouping) different records referring to the same real-world  
entity. Finally, conflicts can arise because of incomplete data,  
erroneous data, and out-of-date data. Returning incorrect data  
in a query result can be misleading and even harmful. This chal-  
lenge is usually addressed by means of data fusion techniques  
that are able to fuse records on the same real-world entity into a  
single record and resolve possible conflicts from different data  
sources.

For instance, if the user is interested in reconstructing the  
current status of orders, then it is necessary to merge the data  
stored in the data source *Batches* and the data stored in *Or-  
ders*. In this case, entity resolution is necessary because the  
same muffin type of the same order is represented by different  
*oids*, e.g., cf. *oid*<sub>101</sub> with *oid*<sub>80</sub> or *oid*<sub>75</sub> with *oid*<sub>70</sub>); data fusion  
is necessary as, when the information about the same muffin  
type in the same order are grouped together, there will be two  
edge symbols, i.e., #, with different semantics, one representing  
the number of ordered pieces and the other one the number of  
cooked pieces.

Traditional approaches that address data heterogeneity pro-  
pose to first solve schema heterogeneity by setting up a data  
integration component that offers a uniform interface to the  
set of data sources. This requires the specification of schema  
mappings that is a really time- and resource-consuming task  
entrusted to data curation specialists. This solution has been  
recognized as a critical bottleneck in large scale deeply het-  
erogeneous and dynamic integration scenarios, as digital fac-  
tories are. A novel approach suggests that mapping creation  
and refinement are interactively driven by the information ac-  
cess needs of service flows and the exclusive role of mappings  
is to contribute to execute service compositions [39]. Hence,  
we start from a chain of services together with their information  
needs expressed as inputs and outputs which we attempt to sat-  
isfy in the dataspace. We may need to discover new mappings  
and refine existing mappings induced by composition require-  
ments, to expose the user to the inputs and outputs thereby dis-  
covered for their feedback and possibly continued adjustments.  
Therefore, the service composition induces a data space orches-  
tration that aims at aligning the data space to the specific service  
goals through the interactive execution of three steps: mapping  
discovery and selection, service composition simulation, feed-  
back analysis. Mappings that are the outcome of this process  
can be stored and reused when solving similar service compo-  
sition tasks.

Essentially, the data flow indicates that from each QRCode  
returned by the *overcook* service, (i) it should be derived the  
Web service to interact with the delivery agent/courier, whereas  
(ii) from the type of the overcooked muffin it should be derived  
the list of ingredients together with the required quantities as  
input to the dosing machine.

Therefore, mapping discovery leads to two mappings

768 whose targets are (QRCode, call, ?z) and (type, 819  
769 has\_ingredient, ?h), (?h, name, ?x), (?h, qty, 820  
770 num\*?y). A plausible output to the mapping discovery for 821  
771 the second mapping is shown in Figure 8. This mapping 822  
772 involves the *Recipes* data source, only, and provides all the 823  
773 ingredients of the recipe of the type of the given overcooked 824  
774 muffins. If some muffins of type `type_1` are overcooked then 825  
775 `?k = type_1` and the inputs to the dosing machine will be 826  
776 (yoghurt, 75gr), (blueberry, 30gr), (egg, 2), etc. Notice 827  
777 that the discovery of such a mapping most likely needs human 828  
778 intervention because, given a muffin type, some alternatives 829  
779 are available to get to the corresponding ingredients and the 830  
780 addition of the basic recipe ingredients is not so obvious. 831

#### 781 4. Discussion and closing remarks 832

782 In this paper, we have proposed an architectural framework 835  
783 for RAMI 4.0-based digital factories. The framework sup- 836  
784 ports agile supply-chains through innovative technological ap- 837  
785 proaches aiming at the dynamic discovery of service and data 838  
786 flows that best fit the requirements expressed in business pro- 839  
787 cess specifications and their evolution. 840

788 The proposed approach relies on a three-level architecture 841  
789 whose aims are to enable the interoperability among the differ- 842  
790 ent parts of the real factory and to ease the involvement of hu- 843  
791 mans in the agile management of factory processes. Moreover, 844  
792 the proposed approach leverages the interactions with other ac- 845  
793 tors of the supply chain, making them easier and overcoming 846  
794 the obstacles deriving from the possible different data formats 847  
795 and process management. 848

796 We now discuss factors that may call the results of the work 849  
797 conducted in this paper into question or diminish the meaning- 850  
798 fulness of the results. These factors are denoted as *threats to* 851  
799 *validity*. 852

800 A first threat to validity is the possible dimension of digital 853  
801 factories which may adopt the proposed approach. An architec- 854  
802 tural approach as the one proposed here, may appear as more 855  
803 suitable to large and “traditional” manufacturing factories (e.g., 856  
804 mechanical, automotive, etc.) than to small factories. Indeed, 857  
805 as pointed out also by the EU Commission in its initiative about 858  
806 *digital transformation*<sup>4</sup>, the most benefits from digital factories 859  
807 will be shown in small realities and in scenarios not fully auto- 860  
808 mated, as the food industry. For this reason, we have presented 861  
809 as a case study the MyMuffin example, in order to make it evi- 862  
810 dent how to successfully apply the approach in other scenarios 863  
811 than manufacturing. 864

812 Another threat is the lack of an extensive validation of the ap- 865  
813 proach, which need to be concretely evaluated in many different 866  
814 situations. In order to diminish this, (i) we have carefully de- 867  
815 signed the approach by integrating best approaches in the differ- 868  
816 ent disciplines, and (ii) by extensively considering the existing 869  
817 state-of-the-art (cf. Section 2) in order to include pros and cons 870  
818 of each proposals, and finally (iii) we have presented a carefully 871

designed and detailed case study in order to make itself an ini-  
tial validation. But undoubtedly, more work is needed in order  
to validate the approach in several different digital factories in  
different business segments and activity areas.

Related to the above threats, there is the question about the  
repeatability of the approach. By considering, as a case study,  
not a traditional manufacturing scenario but a different one, as  
the food industry, we argue that the approach is enough general  
to be applied in many other scenarios, not only the ones for  
which it has been conceived.

Finally, the lack of a software implementation is a crucial  
threat to validity. On this point, we are currently working on  
realizing all the layers of the proposed architecture, on the basis  
of available research prototypes and new ones to be developed  
ad hoc for this research.

Our future work and next steps will mainly consist in ad-  
dressing the above mentioned threats to validity, that is in the  
implementation of the proposed architectural framework and  
proof-of-concept of such an architecture, to be validated in agile  
supply-chain application scenarios. It means to further in-  
vestigate several interesting research issues towards the imple-  
mentation of a polystore with the defined characteristics, the  
dynamic and interactive discovery of data sources and services,  
the dynamic choreography of processes, services and data for  
supply-chain responsiveness.

Finally, we would like to remark the impact of our research,  
which is manifold: from the business to the technological  
facets. Being our approach able to make the supply chains more  
agile, the adoption of the proposed solution will have a con-  
crete impact on the industrial landscape, where companies are  
in need of making their supply chain more agile, but often lack  
proper information systems able to combine the business con-  
straints and opportunities and the ICT potential. As an example,  
only in Italy more than 388 000 Italian manufacturing compa-  
nies are micro, small and medium size enterprises (SMEs, up  
to 249 employees), and they represent 99.7% of the total num-  
ber of manufacturing companies and more than 60% of the total  
turnover (Eurostat 2018). Thus, in many cases, supply chains  
are not driven by big companies, which could provide a sort of  
stability in the relationships among the partners and foster the  
adoption of a common ICT infrastructure. Conversely, Italian  
supply chains (and this is true in many other countries) are very  
fragmented and dynamic, to properly satisfy the multiple differ-  
ent customers requesting tailored products and services. SMEs  
often face the challenge to interact digitally with their counter-  
parts, lacking both standards and resources. Using a common  
reference architecture and an agile ICT infrastructure, our re-  
search offers a solution for these numerous enterprises by en-  
abling the creation of a “co-opetitive” environment where com-  
panies can respond more quickly to the emerging needs of the  
market. The adoption of the proposed reference architecture  
would have also a significant impact on the Italian digital mar-  
ket, whose value was more than € 66 billion in 2016 and grow-  
ing, employing approximately 740 000 people, as companies  
will require to revise their information systems to make them  
compliant to the proposed reference architecture. Although this  
could be seen as a cost that the companies have to bear, on the

<sup>4</sup>Cf. [https://ec.europa.eu/growth/industry/policy/digital-transformation\\_en](https://ec.europa.eu/growth/industry/policy/digital-transformation_en).

876 other side, the opportunity to simplify the participation to the  
877 supply chains and the consequent benefits will definitely com-  
878 pensate the effort.

879 It is worth to mention that the need for agile supply chains  
880 is clear also at European level, as the EFFRA association iden-  
881 tifies “agile value networks” as one of the five key priorities  
882 for the Future of Factories to deliver innovative products with  
883 a high degree of personalization. Thus, the adoption of our  
884 proposal goes into this direction, supporting the achievement  
885 of this goal on a continental scale, in particular considering  
886 that Italian firms strongly interact with European customers and  
887 suppliers. Moreover, most European countries, are similar to  
888 Italy, i.e., with very few large companies that represent a limited  
889 share of the total turnover. At European level SMEs (including  
890 micro companies) represent 99.2% of manufacturing compa-  
891 nies (Eurostat 2018), although with some differences among  
892 countries. Germany for example has a higher share of large  
893 firms (2.1% of the total, representing 74.5% of the turnover,  
894 compared to a European average of 62.8%). For this reason,  
895 our proposal also contribute to the digital transformation of  
896 SMEs all over Europe. At the business level, the impact of  
897 our research is clear also from the customer standpoint, as fa-  
898 cilitating the information exchange and the possibility to react  
899 to negative as well as positive changes occurring in the supply  
900 chains can provide more customized products as well as added  
901 value services to the customer. This is very relevant nowadays,  
902 since *servitization* is more and more pursued by companies to  
903 attract an increasing number of customers that want both cus-  
904 tomized products and services in addition to the products they  
905 buy. Moreover, agility considers security flaws among the po-  
906 tential risks against which supply chains need to be responsive.  
907 Therefore, the adoption of the proposed solution has a funda-  
908 mental value today, considering that security and privacy of  
909 data are one of the most important issues for the public.

910 *Acknowledgements..* This work has been partly supported by  
911 the European Commission through the H2020 project FIRST –  
912 virtual Factories: Interoperation suppoRting buSiness innova-  
913 Tion (grant agreement # 734599).

## 914 References

915 [1] N. Chungoora, R. I. Young, G. Gunendran, C. Palmer, Z. Usman, N. A.  
916 Anjum, A.-F. Cutting-Decelle, J. A. Harding, K. Case, A model-driven  
917 ontology approach for manufacturing system interoperability and knowl-  
918 edge sharing, *Computers in Industry* 64 (4) (2013) 392–401.  
919 [2] M. P. Papazoglou, Smart connected digital factories - unleashing the  
920 power of industry 4.0 and the industrial internet, in: *Proceedings of*  
921 *the 8th International Conference on Cloud Computing and Services Sci-*  
922 *ence, CLOSER 2018, Funchal, Madeira, Portugal, March 19-21, 2018.*  
923 *SciTePress*, 2018, pp. 260–271.  
924 [3] F. Zezulka, P. Marcon, I. Vesely, O. Sajdl, Industry 4.0 an introduction in  
925 the phenomenon, *IFAC-PapersOnLine* 49 (25) (2016) 8 – 12, 14th IFAC  
926 *Conference on Programmable Devices and Embedded Systems PDES*,  
927 2016.  
928 [4] E. F. of the Future Research Association, *Factories 4.0 and beyond*, Tech  
929 *rep.*, EU (2016).  
930 [5] M. Yamamoto, H. Sakamoto, Fdt/dtm framework for field device inte-  
931 gration, in: *2008 SICE Annual Conference, 2008*, pp. 925–928. doi:  
932 10.1109/SICE.2008.4654787.

[6] H. Cai, L. Da Xu, B. Xu, C. Xie, S. Qin, L. Jiang, Iot-based config-  
urable information service platform for product lifecycle management,  
*IEEE Transactions on Industrial Informatics* 10 (2) (2014) 1558–1567.  
[7] F. Tao, Y. Zuo, L. Da Xu, L. Zhang, Iot-based intelligent perception  
and access of manufacturing resource toward cloud manufacturing, *IEEE*  
*Transactions on Industrial Informatics* 10 (2) (2014) 1547–1557.  
[8] Y. Lu, Industry 4.0: A survey on technologies, applications and open  
research issues, *Journal of Industrial Information Integration* 6 (2017) 1  
– 10. doi:<https://doi.org/10.1016/j.jii.2017.04.005>.  
URL <http://www.sciencedirect.com/science/article/pii/S2452414X17300043>  
[9] M. Dumas, M. La Rosa, J. Mendling, H. A. Reijers, *Fundamentals of*  
*Business Process Management*, Second Edition, Springer, 2018. doi:  
10.1007/978-3-662-56509-4.  
[10] W. M. P. van der Aalst, *Process Mining - Data Science in Action*, Second  
Edition, Springer, 2016. doi:10.1007/978-3-662-49851-4.  
[11] C. Janiesch, A. Koschmider, M. Mecella, B. Weber, A. Burattin, C. Di Ci-  
ccio, A. Gal, U. Kannengiesser, F. Mannhardt, J. Mendling, A. Ober-  
weis, M. Reichert, S. Rinderle-Ma, W. Song, J. Su, V. Torres, M. Wei-  
dlich, M. Weske, L. Zhang, The internet-of-things meets business process  
management: Mutual benefits and challenges, *CoRR abs/1709.03628*.  
arXiv:1709.03628.  
URL <http://arxiv.org/abs/1709.03628>  
[12] A. Garcia-Dominguez, M. Marcos, I. Medina, A comparison of  
bpmn 2.0 with other notations for manufacturing processes, *AIP*  
*Conference Proceedings* 1431 (1) (2012) 593–600. arXiv:  
<https://aip.scitation.org/doi/pdf/10.1063/1.4707613>,  
doi:10.1063/1.4707613.  
URL <https://aip.scitation.org/doi/abs/10.1063/1.4707613>  
[13] S. Zor, D. Schumm, F. Leymann, A proposal of bpmn extensions for the  
manufacturing domain, in: *Proceedings of the 44th CIRP International*  
*Conference on Manufacturing Systems*, 2011.  
[14] A. Marrella, M. Mecella, S. Sardiña, Supporting adaptiveness of cyber-  
physical processes through action-based formalisms, *AI Commun.* 31 (1)  
(2018) 47–74. doi:10.3233/AIC-170748.  
[15] T. Erl, *Service-Oriented Architecture: Concepts, Technology, and De-*  
*sign*, Prentice Hall Professional Technical Reference, Upper Saddle River,  
NJ, 2005.  
[16] T. Bangemann, M. Riedl, M. Thron, C. Diedrich, Integration of classical  
components into industrial cyber physical systems, *Proceedings of the*  
*IEEE* 104 (5) (2016) 947–959. doi:10.1109/JPROC.2015.2510981.  
[17] M. Wagner, D. Zbel, A. Meroth, Soda: Service-oriented architecture for  
runtime adaptive driver assistance systems, in: *2014 IEEE 17th Inter-*  
*national Symposium on Object/Component/Service-Oriented Real-Time*  
*Distributed Computing*, 2014, pp. 150–157. doi:10.1109/ISORC.  
2014.15.  
[18] H. Bohn, A. Bobek, F. Golatowski, Sirena - service infrastructure for  
real-time embedded networked devices: A service oriented framework  
for different domains, in: *International Conference on Networking, In-*  
*ternational Conference on Systems and International Conference on Mo-*  
*bile Communications and Learning Technologies (ICNICONSMCL'06)*,  
2006, pp. 43–43. doi:10.1109/ICNICONSMCL.2006.196.  
[19] A. W. Colombo, S. Karnouskos, O. Kaynak, Y. Shi, S. Yin, Indus-  
trial cyberphysical systems: A backbone of the fourth industrial revolu-  
tion, *IEEE Industrial Electronics Magazine* 11 (1) (2017) 6–16. doi:  
10.1109/MIE.2017.2648857.  
[20] E. Tello-Leal, O. Chiotti, P. D. Villarreal, Software agent architecture for  
managing inter-organizational collaborations, *Journal of applied research*  
*and technology* 12 (3) (2014) 514–526.  
[21] B. I. Zhang, Service-oriented logistics supply chain information manage-  
ment system, in: *2018 International Conference on Intelligent Trans-*  
*portation, Big Data Smart City (ICITBS)*, 2018, pp. 428–431. doi:  
10.1109/ICITBS.2018.00114.  
[22] T. Sakakura, A speculation on a framework that provides highly orga-  
nized services for manufacturing, in: *2015 IEEE International Conference*  
*on Automation Science and Engineering (CASE)*, 2015, pp. 1025–1028.  
doi:10.1109/CoASE.2015.7294233.  
[23] G. Castelli, M. Mamei, A. Rosi, F. Zambonelli, Engineering pervasive  
service ecosystems: The sapere approach, *ACM Trans. Auton. Adapt.*  
*Syst.* 10 (1) (2015) 1:1–1:27. doi:10.1145/2700321.

- 1004 URL <http://doi.acm.org/10.1145/2700321>
- 1005 [24] W. He, L. Da Xu, Integration of distributed enterprise applications: A  
1006 survey, *IEEE Transactions on Industrial Informatics* 10 (1) (2014) 35–42.
- 1007 [25] C. Schroth, T. Janner, V. Hoyer, Strategies for cross-organizational ser-  
1008 vice composition, in: 2008 International MCETECH Conference on  
1009 e-Technologies (mcetech 2008), 2008, pp. 93–103. doi:10.1109/  
1010 MCETECH.2008.13.
- 1011 [26] B. Liu, Z. Zhang, Qos-aware service composition for cloud manufactur-  
1012 ing based on the optimal construction of synergistic elementary service  
1013 groups, *The International Journal of Advanced Manufacturing Technol-*  
1014 *ogy* 88 (9) (2017) 2757–2771. doi:10.1007/s00170-016-8992-7.  
1015 URL <https://doi.org/10.1007/s00170-016-8992-7>
- 1016 [27] M. B. Karimi, A. Isazadeh, A. M. Rahmani, Qos-aware service com-  
1017 position in cloud computing using data mining techniques and genetic  
1018 algorithm, *J. Supercomput.* 73 (4) (2017) 1387–1415. doi:10.1007/  
1019 s11227-016-1814-8.  
1020 URL <https://doi.org/10.1007/s11227-016-1814-8>
- 1021 [28] M. Gunzert, Compatibility and interoperability in field device integration;  
1022 a view on eddl, fdt and fdi, in: 2015 54th Annual Conference of the Soci-  
1023 ety of Instrument and Control Engineers of Japan (SICE), 2015, pp.  
1024 941–946. doi:10.1109/SICE.2015.7285561.
- 1025 [29] M. Hepp, J. Leukel, V. Schmitz, A quantitative analysis of ecl@ss, un-  
1026 spsc, eotd, and rnd content, coverage, and maintenance, in: IEEE Inter-  
1027 national Conference on e-Business Engineering (ICEBE’05), 2005, pp.  
1028 572–581. doi:10.1109/ICEBE.2005.15.
- 1029 [30] G. De Giacomo, D. Lembo, M. Lenzerini, R. Rosati, On reconciling data  
1030 exchange, data integration, and peer data management, in: Proceedings  
1031 of the Twenty-Sixth ACM SIGACT-SIGMOD-SIGART Symposium on  
1032 Principles of Database Systems, June 11-13, 2007, Beijing, China, 2007,  
1033 pp. 133–142.
- 1034 [31] P. G. Kolaitis, Reflections on schema mappings, data exchange, and meta-  
1035 data management, in: Proceedings of the 37th ACM SIGMOD-SIGACT-  
1036 SIGAI Symposium on Principles of Database Systems, Houston, TX,  
1037 USA, June 10-15, 2018, 2018, pp. 107–109.
- 1038 [32] M. J. Franklin, A. Y. Halevy, D. Maier, From databases to dataspace:  
1039 a new abstraction for information management, *SIGMOD Record* 34 (4)  
1040 (2005) 27–33.
- 1041 [33] P. Cudré-Mauroux, M. T. OZSU, Peer Data Management System,  
1042 Springer US, Boston, MA, 2009, pp. 2055–2056.
- 1043 [34] V. Gadepally, P. Chen, J. Duggan, A. J. Elmore, B. Haynes, J. Kepner,  
1044 S. Madden, T. Mattson, M. Stonebraker, The bigdawg polystore system  
1045 and architecture, in: 2016 IEEE High Performance Extreme Computing  
1046 Conference, HPEC 2016, Waltham, MA, USA, September 13-15, 2016,  
1047 2016, pp. 1–6.
- 1048 [35] J. Wang, T. Baker, M. Balazinska, D. Halperin, B. Haynes, B. Howe,  
1049 D. Hutchison, S. Jain, R. Maas, P. Mehta, D. Moritz, B. Myers, J. Ortiz,  
1050 D. Suciu, A. Whitaker, S. Xu, The myria big data management and analyt-  
1051 ics system and cloud services, in: CIDR 2017, 8th Biennial Conference  
1052 on Innovative Data Systems Research, Chaminade, CA, USA, January 8-  
1053 11, 2017, Online Proceedings, 2017.
- 1054 [36] A. Marrella, M. Mecella, B. Pernici, P. Plebani, A design-time data-  
1055 centric maturity model for assessing resilience in multi-party business  
1056 processes, *Information Systems* doi:<https://doi.org/10.1016/j.is.2018.11.002>.
- 1057 [37] A. Marrella, M. Mecella, S. Sardina, Intelligent process adaptation in the  
1058 smartpm system, *ACM Transactions on Intelligent Systems and Technol-*  
1059 *ogy* 8 (2) (2016) 1–43.
- 1060 [38] W. Penzo, S. Lodi, F. Mandreoli, R. Martoglia, S. Sassatelli, Semantic  
1061 peer, here are the neighbors you want!, in: Proceedings of the 11th in-  
1062 ternational conference on Extending database technology: Advances in  
1063 database technology, ACM, 2008, pp. 26–37.
- 1064 [39] F. Mandreoli, A framework for user-driven mapping discovery in rich  
1065 spaces of heterogeneous data, in: OTM Confederated International Con-  
1066 ferences” On the Move to Meaningful Internet Systems”, Springer, 2017,  
1067 pp. 399–417.
- 1068