A Cloud-based Framework for Shop Floor Big Data Management and Elastic Computing Analytics

German Terrazas[a,*], Nicolas Ferry[b], Svetan Ratchev[c]

[a]*Institute for Manufacturing, University of Cambridge, UK*
[b]*SINTEF Digital, Norway*
[c]*Institute for Advanced Manufacturing, University of Nottingham, UK*

Abstract

Advanced digitalization together with the rise of disruptive Internet technologies are key enablers of a fundamental paradigm shift observed in industrial production. This is known as the fourth industrial revolution (Industry 4.0) which proposes the integration of the new generation of ICT solutions for the monitoring, adaptation, simulation, and optimisation of factories. With the democratization of sensors and actuators, factories and machine tools can now be sensorized and the data generated by these devices can be exploited, for instance, to optimise the utilization of the machines as well as their operation and maintenance. However, analyzing the vast amount of generated data is resource demanding both in terms of computing power and network bandwidth, thus requiring highly scalable solutions. This paper presents a novel big data approach and analytics framework for the management and analysis of machine generated data in the cloud. It brings together standard open source technologies and the exploitation of elastic computing, which, as a whole, can be adapted to and deployed on different cloud computing platforms. This enables reducing infrastructure costs, minimizing deployment difficulty and providing on-demand access to a virtually infinite set of computing power, storage and network resources.

Keywords: Industry 4.0, cyber physical systems, big data, cloud-based data collection, cloud-based analytics, elastic computing

1. Introduction

Advanced digitalisation together with information and communication technologies (ICT) are widely recognised for their potential to drive digital transformations in business and industries while enhancing mass production and underpinning product innovation [1][2]. This refers to Industry 4.0, which comprises a range of novel concepts such as smart factory, cyber-physical systems,

*Corresponding author
Email address: gt401@cam.ac.uk (German Terrazas)

self-organisation, adaptation, sustainability and resource-efficiency [3, 4, 5]. Although there is no official classification, the broad consensus indicates that the key Industry 4.0 enabling technologies are industrial IoT, cloud computing, big data and cyber security [6, 7]. The smooth interplay between these is important as it allows the collection, transmission and storage of raw data through cloud-based solutions to, in turn, generate actionable intelligence. In fact, sensors, cyber-physical devices as well as big data technologies and distributed computing infrastructures were brought together in different context [8, 9, 10, 11]. However, their seamless integration remains a challenge as it entails customisation for the manufacturing domain, standardisation, communication architectures as well as control algorithms besides the willingness from manufacturing sites [12, 13]. In addition, existing solutions are mainly vendor locked as they are sold with the machine or are meant to be used only with machines from the same vendor, consequently proving challenging to extend functionality if third party sensors are added to the machines.

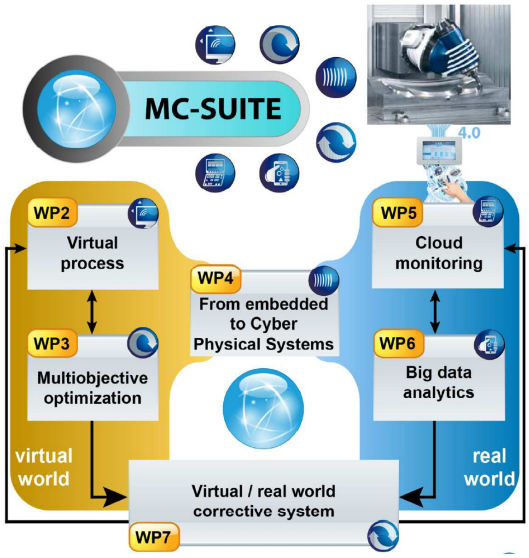


Figure 1: Graphical representation of the MC-SUITE research project and the interaction of its six modules. Image created by IDEKO (`http://www.mc-suite.eu`).

The MC-SUITE project proposes a new generation of ICT-enabled manufacturing process simulation and optimisation that intertwines physical measurements and monitoring, hence transforming the manufacturing industry to dramatically improve product quality and increase yield (see Figure 1). In particular, this paper focuses on open-source technologies integration for realising two of the six MC-SUITE modules: the MC-Monitor and the MC-Analytics. The first one is a data management environment dedicated to collect, pre-process, transmit and store continuously generated machining data. The second one is a cloud-based data analysis framework dedicated to extract

actionable knowledge. Thus, covering the phases of data acquisition, extraction, integration, analysis and interpretation [14]. These novel modules are vendor independent solutions facilitating the connection to any machine or control box with sensors. Although their design is driven by use case requirements and illustrated in the machining context, the interplay between the modules remains generic and can be applied to multiple machine tools. In the following, Section 2 presents current advances on manufacturing data-driven solutions and Section 3 sets the context for this contribution. Section 4 and Section 5 present the design and development of the MC-Monitor and the MC-Analytics respectively. Section 6 presents an enhancement to support distributed big data analytics in the cloud. Finally, Section 7 discusses the presented models and further work while Section 8 summarises the contributions.

2. Related Work

The innovation dynamics, technology evolution and potential impact of IoT technologies have been defined and ranked across four different categories: communication control, network systems, wireless transmission and data processing [15]. Most of these digital innovations are owned by private ICT companies, hence revealing geographically bounded developments, a lack of inter-organisational collaborations and, therefore, making standardisation and interoperability difficult to achieve.

Notwithstanding, there is a plethora of successful research dedicated to improve domain-specific manufacturing processes in terms of data acquisition and analytics. For example, an RFID-enabled intelligent environment for tracking shop floor elements together with two bespoke protocols for management, transmission and warehousing of 30 TB of data per day is presented in [16]. A modularised big data solution built to create prediction models encompassing data specification, storage, processing and analytics for metal cutting is reported in [17]. Although this is a four modules cost-effective approach based on standardised data interfaces, interoperable data exchange formats and open source solutions, it relies on a centralised topology that acts as a single point of failure lacking scalability, loose coupling and independence.

Data acquisition and management architectures built in terms of open source technologies (Apache Kafka[1] and Storm[2]) to address process monitoring, data analysis and fault detection in agricultural harvesters were defined in [18]. Related to this, a cloud-based approach for data collection and processing applied to resource monitoring and adaptive process planning applied to CNC machine tools, sensors and scheduling processes is reported in [19]. More recently, approaches for smart manufacturing using cloud and big data employing open source standard technologies at physical, network and data application levels while promoting the use of private cloud infrastructures were shown in [20][21].

[1] `http://kafka.apache.org`
[2] `http://storm.apache.org`

Although smoothly integrated, the offered solutions are built to meet particular needs without discussing implications in terms of flexibility to extend functionality, e.g. the use of elastic computing models. Moreover, technologies are conveniently hardwired hence missing a higher level of abstraction to enable the generalisation of the approach. More importantly, these solutions say little about advantages for incorporating new sensors or other sources of data.

There is a large body of related research that could be extensively enumerated and discussed [22, 23, 24, 25, 26, 27]. Most of these approaches show well supported arrangements of bespoke applications integrated to off-the-shelf data management frameworks, standardized file formats and product embedded information devices. However, the majority of them lack capabilities for seamless integration between and within already existing tools. Although technological elements to make portability and compatibility smooth do exist, this results in clunky information transfer where potential miscommunication can result in delays and errors. More importantly, cloud computing infrastructures are insufficiently tackled in the literature. This is crucial as they enable ubiquitous information provision, thus playing a key role in the realisation of *"design anywhere, manufacture anywhere"* [28]. Cloud computing together with big data technologies play key roles in managing vast amounts of manufacturing resources providing powerful capabilities for storing, processing and visualisation. Additionally, cloud computing infrastructures offer the capability to deliver both software and hardware resources as services in a highly elastic and scalable way [29]. As explained in [32], there is a plethora of Infrastructure as a Service (IaaS) providers on the market and, unfortunately, the lack of interoperability among these leads to vendor lock-in. This prevents cloud application developers to exploit the peculiarities of existing cloud infrastructure solutions in order to, for example, optimise performance, availability and costs [30, 31]. Hence, configuration and multi-cloud deployment tools, like CloudMF [32], have been developed. The following section sets an industrial context employed for driving the development of the modules. Although this focuses on machining, the resulting models are generic and remain unbound to any specific use case.

3. Industrial Use Case

Monitoring and gaining insight from the energy consumption of machine tools is a major industrial challenge as it has an impact on the overall cost of production. Energy consumption is one of the most important factors of sustainable machining and it enhances the competitiveness of corporations in terms of lower production costs, higher revenue and greener footprint. Therefore, having a better understanding for, ultimately, optimising the energy consumption of machine tools is a challenging task that involves descriptive capabilities, practicability, scalability as well as other relevant efforts to deal with large numbers of components, complexity and variety of machining systems [33].

The Energy Consumption use case was elicited with the industrial partner Soraluce to report the kilowatts per hour consumed by a computer numerical controlled (CNC) machining centre when running a part program during a given

period of time. Although this is a specific problem, the large data sets considered here as well as the overall goal drive the design, development and demonstration of our solution. Due to the complex structure of a CNC machining centre and, since the interest of the use case provider is to equip operators and line managers with an analytics dashboard for diagnostic and maintenance, the utilised profile considers controllable factors selected by Soraluce. Thus, the energy consumption of a machining process is seen as an aggregation of several energy values captured across machining phases like the actual cutting, material feed, spindle exchange, tool exchange, etc. In particular, these analytics involve calculating the energy and time consumption with respect to part programs, spindle heads, machining tools and machine motors. From the operational point of view, only one part program can be running in a CNC machine at a given period of time. When the machining process is taking place, the spindle head can operate within one of the following types: Direct, Automatic or Orthogonal. For each spindle head, only one tool can be fixed – referred by unique tool number – where in particular, tool number zero (Tool 0) denotes the case when the spindle head is empty. In the case of Soraluce CNC machining centres, the energy consumption involves measurements acquired from a set of specific sensors listed in Table 1.

Table 1: The set of specific sensors embedded to Soraluce CNC machining centres. Example values at https://doi.org/10.6084/m9.figshare.5554843.v1

Sensor Name	Description
$Cnc\_Program\_Name\_RT$	Name of the part program
$Cnc\_Program\_BlockNumber\_RT$	Line number of the part program in execution
$Cnc\_Tool\_Number\_RT$	Socket number a tool has been picked up from
$Cnc\_IsCycleOn\_RT$	If part program executing and tool cutting material
$Cnc\_IsAutomaticModeActive$	If auto mode is selected and part program is running
$Cnc\_IsManualModeActive$	If operator is at the machine
$Spindle\_IsAutomatic$	If CNC operating in Automatic spindle mode
$Spindle\_IsDirect$	If machine is operating in Direct spindle mode
$Spindle\_IsOrthogonal$	If machine is operating in Orthogonal spindle mode
$Spindle\_Power\_percent$	Spindle power as percentage of the max. value
$Spindle\_speedActual\_rpm\_d1$	Speed of the machine spindle
$Axis\_FeedRate\_actual$	Speed between the part and the cutting tool
$Axis\_X\_positionActualMCS\_mm\_d1000$	Stage X-axis pos. in Machine Coordinate System (MCS)
$Axis\_X\_positionActualWCS\_mm\_d1000$	Stage X-axis pos. in Workpiece Coordinate System (WCS)
$Axis\_Y\_positionActualMCS\_mm\_d1000$	Stage Y-axis pos. of the stage in the MCS
$Axis\_Y\_positionActualWCS\_mm\_d1000$	Stage Y-axis pos. of the stage in the WCS
$Axis\_Z\_positionActualMCS\_mm\_d1000$	Stage Z-axis pos. of the stage in the MCS
$Axis\_Z\_positionActualWCS\_mm\_d1000$	Stage Z-axis pos. of the stage in the WCS
$Axis\_X1\_power\_percent$	Power as percentage of the maximum value
$Axis\_X2\_power\_percent$	Power as percentage of the maximum value
$Axis\_Y\_power\_percent$	Power as percentage of the maximum value
$Axis\_Z\_power\_percent$	Power as percentage of the maximum value

4. Data Management Environment

4.1. Data Characterisation

Soraluce CNC machining centres are embedded with a large variety of sensors, the values of which are read, approximately, every second. These sensors capture machining conditions (measurements) such as spindle rates, feed rates, part programs, power consumption, block numbers, alarms and operators annotations to name a few. Additionally, some machines are equipped with accelerometres and acoustic emission sensors as well as video and audio devices (i.e. sensors that are not directly related to a CNC machine) for capturing vibration, plastic deformation, and streaming image and sound of the processes being conducted. The type of generated data is called thin data because it is a very little amount of information per device (blip of information) but potentially thousands of devices being polled on a frequent rate. In a previous work, the data generated by Soraluce CNC machines was characterised in terms of variety, velocity, volume and veracity [34]. This characterisation revealed that, due to high velocity, large volume, and heterogeneity data, traditional data management and data processing applications result inadequate for revealing insight. In order to address these, the MC-Monitor and the MC-Analytics modules were designed and developed as independent cloud-based solutions distributed across the physical, network and cyber level as shown in Figure 2. This set up ensures optimal ingestion and transfer of data as well as the availability of and the access to the right data. Moreover, it provides a solution that can extend legacy and under-equipped (in term of sensors) CNC machines while enabling systematic data access and, consequently, cost-effective knowledge extraction.

4.2. Shop Floor Data Management

Data management can be divided into (a) gathering the data from the shop floor, and (b) pre-processing and accessing it. The data generated by Soraluce machines is read by an advanced monitoring system called Savvy Smart Box[3]. This system retrieves, packs and transmits the sensory data to a cloud-based platform called Savvy Industrial Cloud[4] via its machine-to-cloud protocol and makes data available through a REpresentational State Transfer (REST) API. At this point, MC-Monitor fetches, pre-processes and provides access to the collected data by either storing or keeping it in motion. Thus, the main software component of MC-Monitor is a stream processing engine embedded with services to clean and pre-process data. The most prominent engines are Apache Storm, Spark, Flink [5] and Heron [6] all of which rely on similar concepts: data source, event, data stream, event processing and data flow.

[3] http://www.savvydatasystems.com/advanced-monitoring-2

[4] http://solucionestic.conetic.info/cont_ind_conectada/savvy-industrial-cloud-m2c-m2m-solution/

[5] https://flink.apache.org

[6] https://apache.github.io/incubator-heron

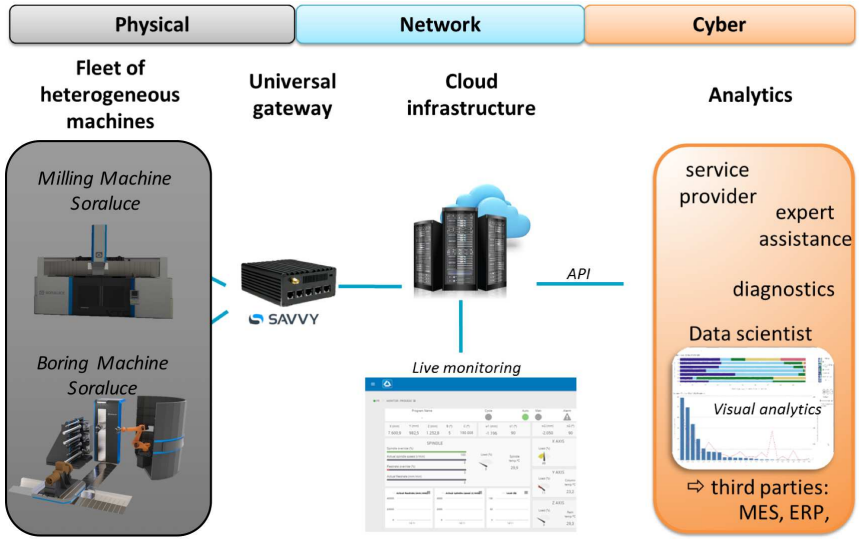


Figure 2: The Savvy gateway serves as a sensor manager from where shop floor data is fetched and transmitted to the MC-Monitor. The MC-Analytics utilises this data to provide analytics and expert assistance to the end user. Image by MC-SUITE newsletter 2.

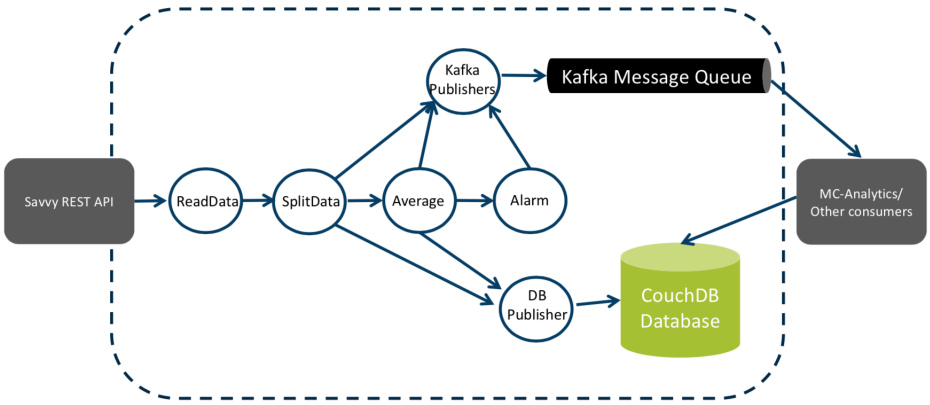


Figure 3: Overview of the MC-Monitor Apache Storm acyclic topology. The *ReadData* spout collects data from the Savvy REST API which is then passed to the *SplitData* bolt. This splits and transmits the data in tuples to the *KafkaPublisher*, *DBPublisher* and *Average* bolts. The *Alarm* bolt works together with the *Average* bolt to alert when measurements fall outside average. The implementation is available at `https://github.com/nicolasferry/vsepml`.

The Heron framework seems to be the most recent solution for processing streams of data, however, the community around it is still modest compared to the other frameworks. Hence, in order to facilitate a future migration to Heron, the MC-Monitor relies on Apache Storm which delivers high performance and backward compatibility to Heron ensuring seamless migration of data processing

topologies. In particular, the logic of the MC-Monitor is specified in terms of an acyclic graph topology where nodes represent sources of streams (spouts) or data processing components (bolts), and the edges represent streams of data (see Figure 3). This solution is scalable since the execution of spouts and bolts can be parallelized and the framework guarantees that data is processed once. Additionally, the computing infrastructure hosting the storm topology can be extended with new virtual machines.

4.2.1. Offline Data Management

This feature prevents network congestion caused by the transmission of large data sets. Thus, a cloud-based solution is employed to store use case specific shop floor data published by the *DBPublisher* bolt as well as messages (video, audio or text) from the operator. Document databases are used for the offline data management since they are attractive for cloud-based applications where speed of deployment is an important issue [35, 36]. Thus, CouchDB[7] was chosen since it is schema-less, supports structured as well as unstructured data, it is horizontally scalable and it exposes a native REST interface. Relevant CNC machine attributes are captured by individual JSON documents (i.e. name, identification and shop floor location) and stored in a database called `MachinesList`. Data gathered at a given point in time is captured in a single JSON document structured as a list of individual measurements and stored in a single database named after the machine it belongs to. Each measurement comprises a sensor identifier, value of the sensor reading, type of information, unit of the value and a coefficient (see example in Listing 1). This document may also contain an extra field called `DocumentSkipped` generated by a sparsity mechanism [34].

Listing 1: A machine sensory data stored in a CouchDB database where *_id* encodes in EPOCH the time sensors were read.


```
{
  "_id":"1451692802000",
  "_rev":"1-ca46f3b012d07e31ed777bc97fa95863",
  "DocumentSkipped":891,
  "Measurements":
  [
  { "Measurement": "335",
    "SensorID": "Axis_FeedRate",
    "Type": "actual",
    "Unit": "",
    "Coeff": "" },
  { "Measurement": "PROGRAM_NAME.H",
    "SensorID": "Cnc_Program_Name",
    "Type": "RT",
    "Unit": "",
    "Coeff": "" },
  { "Measurement": "50",
    "SensorID": "Spindle_Power",
```


[7] http://couchdb.apache.org

```
    "Type": "percent",
    "Unit": "",
    "Coeff": "" },
  { "Measurement": "300",
    "SensorID": "Cnc_Tool_Number",
    "Type": "RT",
    "Unit": "",
    "Coeff": "" }
  ]
}
```

4.2.2. Online Data Management

This feature is built to provide instant and responsive data streams processing while maximising benefit gain from big data. From a high level classification point of view, data streams can be categorised as either transactional or measurement. Shop floor data fall within the last category and the challenges posed for implementing online streaming features like this impose certain architectural and functional requirements. The open source message queue technology Apache Kafka was chosen as it offers performance in terms of message publication and consumption, it is horizontally scalable and fault tolerant, it provides space and time decoupling, and it exposes a simple REST interface with the capability to navigate and re-read messages. Thus, the *KafkaPublisher* bolt (see Figure 3) is capable to publish sensory data into one or more topic (queues where one or multiple subscribers can register to listen and read data of interest). Topics can be created dynamically facilitating scalability and flexibility between the MC-Monitor and the MC-Analytics modules. Currently, the **SensorsChunk**, **Sensor**, **Alarms** and **Average** topics have been created. More details of implementation such as the description of the topics and the structure of the messages can be found in [34].

5. Cloud-based Data Analytics Framework

5.1. Architecture

The conceptual architecture for MC-Analytics is defined as a set of *layers*. A layer, in this context, is a logical division that groups software components by functionality without taking into account their physical location. Layers can be seen as elements arranged on top of each other where the fundamental concept is the isolation, i.e. software components within a layer share common functionally, they are independent from those located in other layers and have no knowledge of their internal structure. Also, each layer is restricted to communicate with the layer above or the layer below, and is allowed to invoke functionality from the lower adjacent layer only. Using a layered scheme brings the following benefits:

- Independence: it enables understanding a single layer as a coherent whole without knowing much about the other layers.
- Loose-coupling and modularity: it facilitates the substitution of layers with alternative implementations of the same basic services.

- Reusability: the software components within a layer can be used as building blocks to build and deliver higher-level services.

Additionally, this logical separation brings distribution, flexibility and scalability to the proposed framework. Although the number of layers depends on the system complexity, it is well known that most approaches comprise four layers [37] which is in line with the MC-Analytics architecture shown in Figure 4.

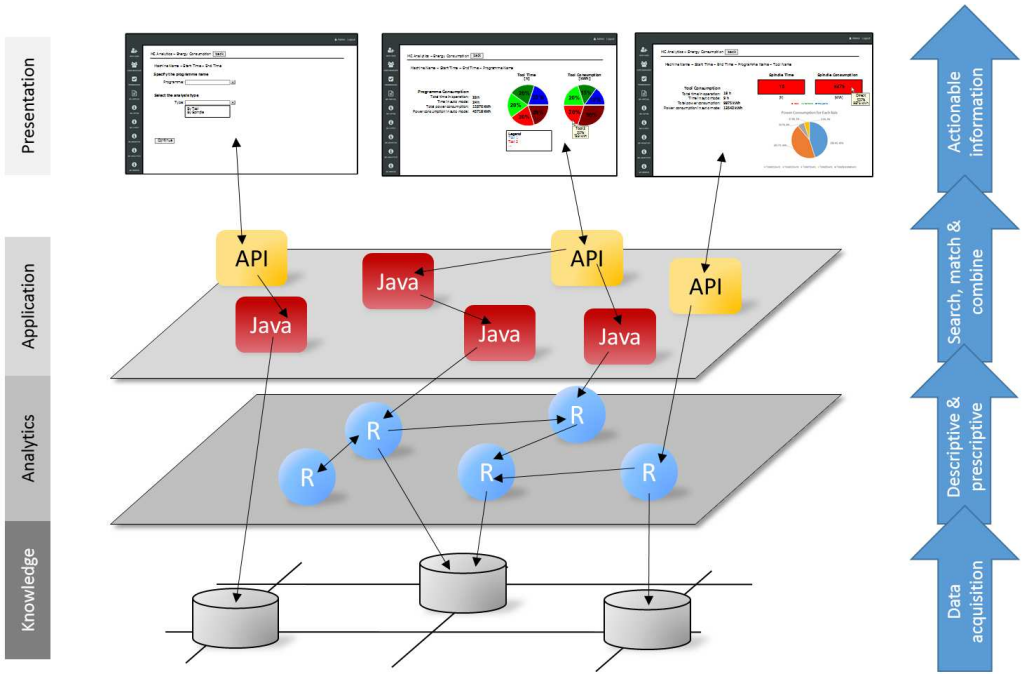


Figure 4: The MC-Analytics conceptual architecture comprising different levels of abstraction. The *Knowledge* layer captures relevant sources of manufacturing data accessed by knowledge extraction tasks at the *A*nalytics layer. The effective combination of these is done at the *Application* layer which is also responsible for handling end users' request and the return of actionable information to the *P*resentation layer.

5.2. Data Analytics Framework Layers

5.2.1. Knowledge Layer

The *Knowledge* layer comprises online and offline data sources for storage and retrieval of persistent manufacturing data. Therefore, its implementation utilises the MC-Monitor module seen in Section 4.2.

5.2.2. Analytics Layer

This layer defines the environment where systematic knowledge extraction takes place. Although there is no specific technology for this level, the main requirement is to deploy software components that implement the Service Layer

pattern [38]. To illustrate this, R language was chosen since it offers capabilities to develop from very simple descriptive analytics to complex prescriptive ones, it supports multicore task distribution and, more importantly, it is a popular tool in numerical analysis and machine learning. Five independent bespoke R scripts were developed to deliver the descriptive analytics required for the Energy Consumption use case: **AllToolsAnalytics.R** and **AllSpindlesAnalytics.R** that return the energy consumption with respect to the tools or spindle types; **MotorsPerToolAnalytics.R** and **ToolsPerSpindleAnalytics.R** that return energy consumption of the motors with respect to a specific tool or a specific spindle type respectively; and **MotorsPerToolInSpindleAnalytics.R** that returns the energy consumption with respect to the motors when using a specific spindle type and tool. Since these scripts must be triggered from the *Application* layer, the solution of choice is Rserve[8] which facilitates remote invocation over a network while offering a wide range of compatible clients. This makes MC-Analytics flexible since it requires no initialisation of R programming environment nor linking the invoker against a particular library.

5.2.3. Application Layer

This layer defines an environment where back end logic and analytics orchestration takes place. The main requirement is to deploy software components that control transactions, search and execute analytics, and coordinate responses while being suitable for remote invocation. These elements of software implement the Command pattern (see Figure 5) as it allows the *Presentation* layer, and in fact any other client, to perform requests. For implementation purposes, Java and Jersey[9] were chosen to support reusability, scalability and a neat linkage to remote access. In the context of the Energy Consumption use case, the Invoker is a class called **EnergyConsumptionREST** that groups all expected functionality. This collaborates with **ActionsManager**, which is a helper class registering Command subclasses for discovery. The Command is implemented as an abstract class called **Action** with subclasses **AllToolsEC**, **MotorsECPerTool**, **AllSpindlesEC**, **ToolsECPerSpindle** and **MotorsECPerToolInSpindle** representing functionalities within the use case. Finally, the **RConnection** is the Receiver located in the *Analytics* layer (see Figure 6).

5.2.4. Presentation Layer

This layer comprises components that manage end user external interactions and display actionable information. Since different use cases could be associated to different types of visualisation, a large range of eligible visualization solutions can be explored. For the Energy Consumption use case, Vaadin[10] was selected for developing the data analytics dashboard. It is important to note that this

[8] https://www.rforge.net/Rserve/
[9] `https://jersey.github.io`
[10] `https://vaadin.com/`

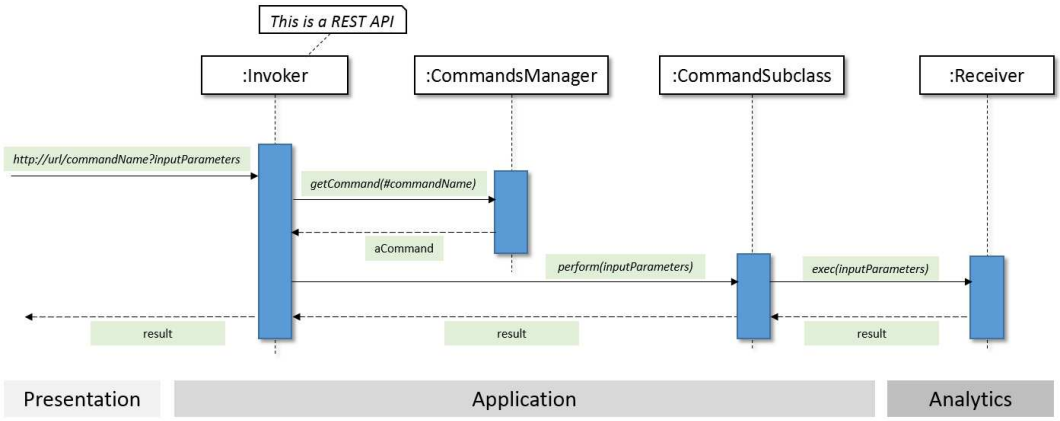


Figure 5: Interaction diagram across the *Presentation*, *Application* and *Analytics* layers. The Invoker triggers analytics by encapsulating the request itself into an object that can be stored and passed around like a standard application object.

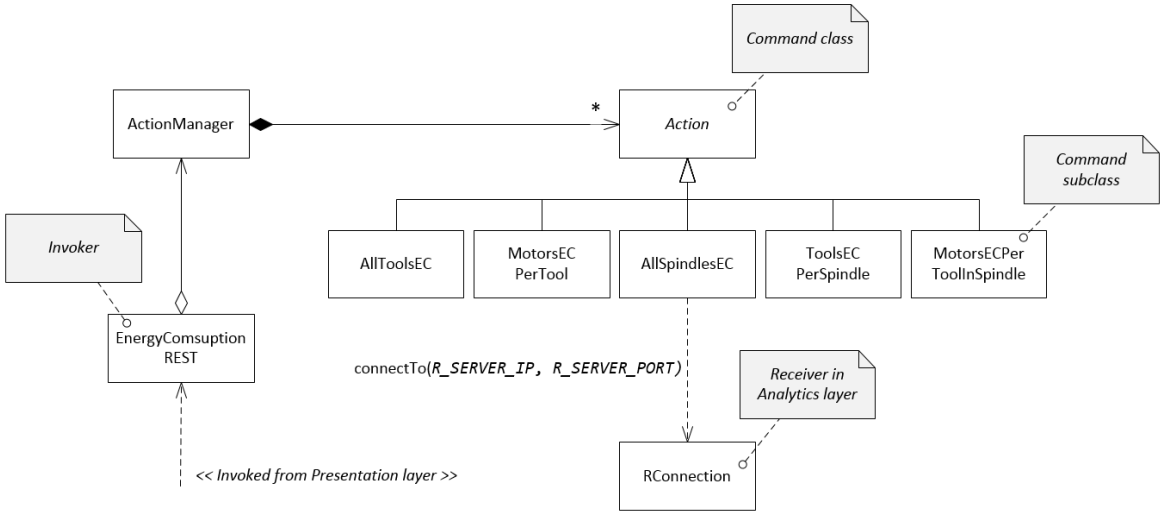
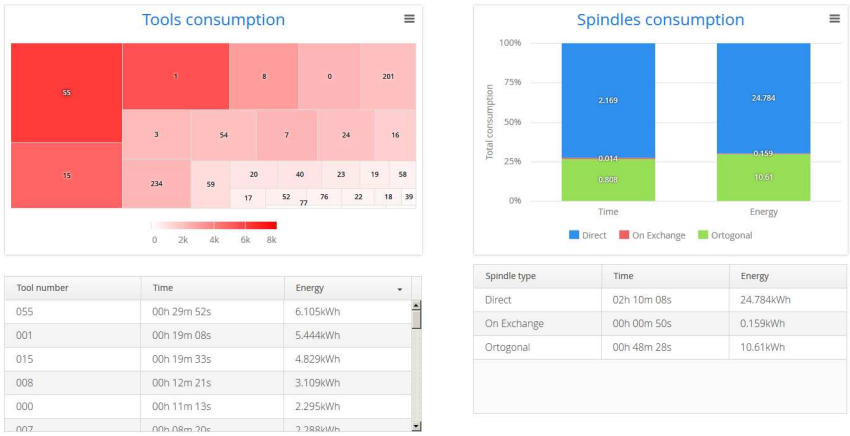


Figure 6: Class diagram of the Energy Consumption use case. The Invoker collaborates with ActionsManager in charge of registering Command subclasses for discovery. Subclasses of Action reflect more particular use case functionalities.

selection does not exclude other solutions, i.e. should further use cases require specific visualization needs, new solutions could be utilized. Figure 7 depicts two energy consumption dashboard examples for a particular part program ran in a specific CNC machine during a given period of time. Each of these is calculated in terms of energy values captured across machining phases like cutting, material feed, spindle exchange, tool exchange, etc. Thus, facilitating visually actionable information that can be used for monitoring, decision making, diagnostic and

maintenance.



(a) (b)

Figure 7: Energy consumption dashboards with respect to (a) the tools and (b) the spindle types. In (a), a tree map classifies energy consumption per tool according to area size (time) and colour shade (KWh). In (b), accumulative plots summarise the percentage of time and energy consumption per type of spindle.

5.3. Cloud Deployment

The Amazon Elastic Compute Cloud (Amazon EC2) was chosen as deployment platform since it facilitates and speeds up implementation. The main feature of this IaaS provider is the variety of machine images, instances, instance types and storage. Amazon Machine Images (AMIs) and instances are central to the Amazon EC2 infrastructure. An AMI is a template that contains an initial software configuration such as the operating system, application servers, and other type of applications. An instance is a concrete occurrence of an AMI and it is always associated to an instance type[11] that, essentially, defines the hardware configuration of the underlying computer where it runs. AWS enables users to create tailored AMIs to quickly and easily start instances customised with everything needed to run applications. The *Analytics* layer is realised with an AMI configured with Ubuntu OS together with the analytic components and Rserve. Therefore, once an instance of this image is launched, Rserve provides remote access to perform analytics. Likewise, the *Application* layer is realised with another AMI configured with Ubuntu OS and Apache Tomcat equipped with all classes described in Section 5.2.3. Once an instance is launched, the application server starts and the invokers ready to accept request via REST API. The *Presentation* layer is developed and deployed in a third AMI whereas

[11] `https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instance-types.html`

the *Knowledge* layer database is available in a remote service part of the MC-Monitor module. Figure 8 summarises the organisation and deployment of the different software components.

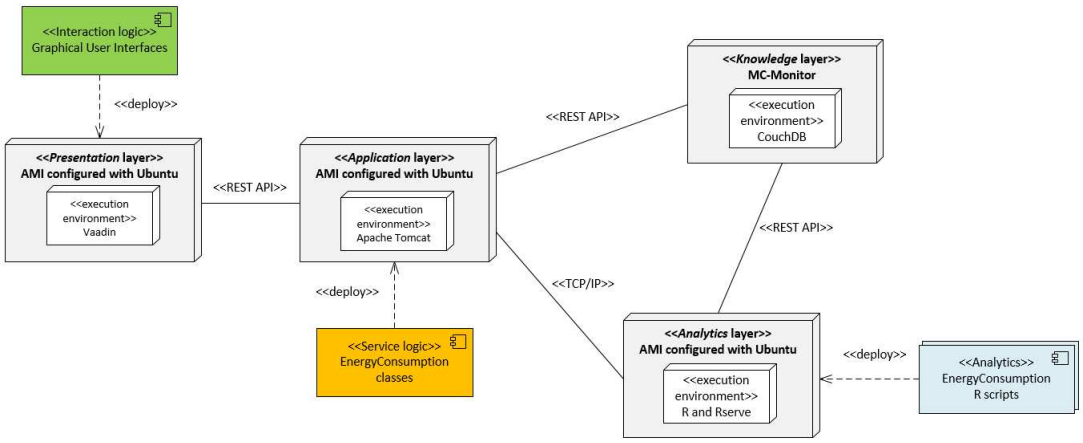


Figure 8: Deployment of MC-Analytics in UML showing the hardware and virtual platforms. The ≪deploy≫ dependency indicates which artifacts are deployed and the ≪execution≫ nodes represent the environments where these execute.

6. Elastic Data Analytics Model

Elastic computing is considered as one of the central elements of the cloud computing paradigm and is defined as the ability to adapt to workload changes by scaling up (provision) and scaling down (deprovision) computing resources in an automatic manner, in such a way that at a given point in time the available computing resources match the current demands [39][40][41]. Employing an elastic computing infrastructure is essential when developing analytics solutions as these would potentially handle large data sets. In fact, one of the well-known limitations of R is efficient memory management as its performance degrades when dealing with large volumes of data. This drawback is experienced when executing scripts developed for Energy Consumption use case over two weeks of data which, in other words, represents more than a million JSON documents of 22 sensors each. A solution to this is leveraging R scripts with distributed computing power deployed over an elastic computing infrastructure. For demonstration purposes, ProActive[12] has been chosen. Thus, the scripts reported in Section 5.2.2 together with arrays of values could be rapidly set as input parameters to the PASolve[13] function which allows parametric sweep, i.e. multiple and asynchronous executions of a script with different input values over

[12] `https://www.activeeon.com/big-data-automation/parallel-r`
[13] `https://try.activeeon.com/tutorials/r/r.html`

a workflow of independent nodes. Since ProActive requires an arrangement of computing resources distributed across and communicated within a network, the Amazon EC2 can be effectively used. Figure 9 depicts the interplay across layers while delivering an evolution from a monolithic to a distributed architecture and computing enhancement for MC-Analytics.

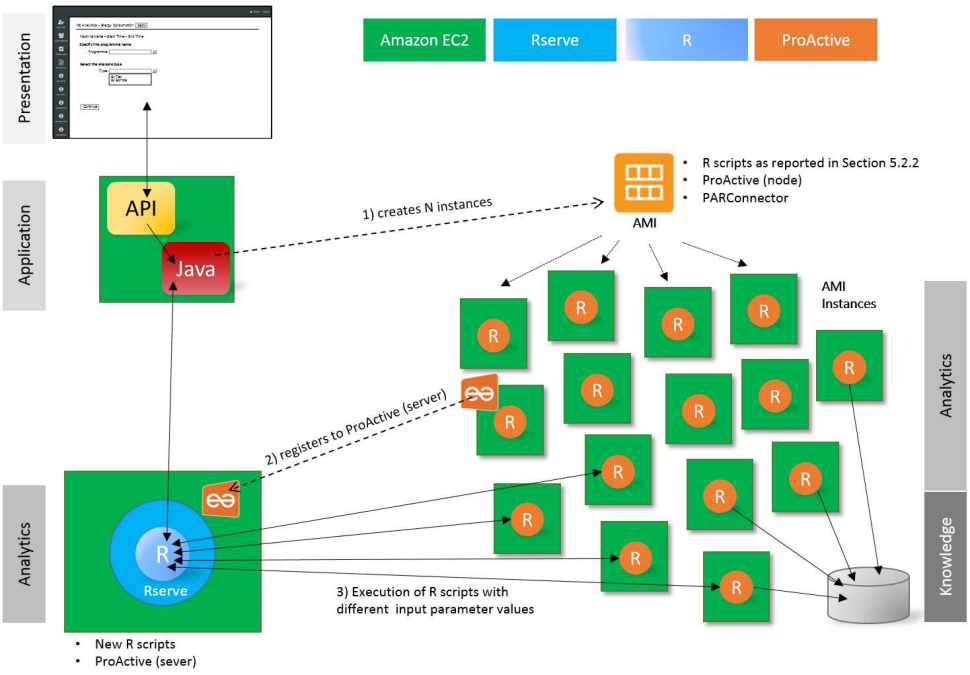


Figure 9: An Action (red) at the *Application* layer creates a number of computing resources (1) which are used by ProActive when an R component (blue) invokes PASolve. This solution performs efficient and fast distributed analytics in the cloud.

6.1. Implementation

Integrating and exploiting elasticity could result cumbersome if a system solution has not been designed for it. Therefore, the independence, loose-coupling and reusability benefits offered by the MC-Analytics architecture are crucial for delivering an implementation. In particular, the *Application* layer was enriched with a new **Action** subclass called **AllToolsECElastic** that works together with an **AmazonEC2Manager** object in charge of programmatically creating, managing and releasing instances of an AMI configured with ProActive nodes. Such elastic computing resource cycle is achieved by providing infrastructure specific values to the Amazon EC2 service via the AWS SDK. The UML diagram presented in Figure 10 depicts the relationship between the classes.

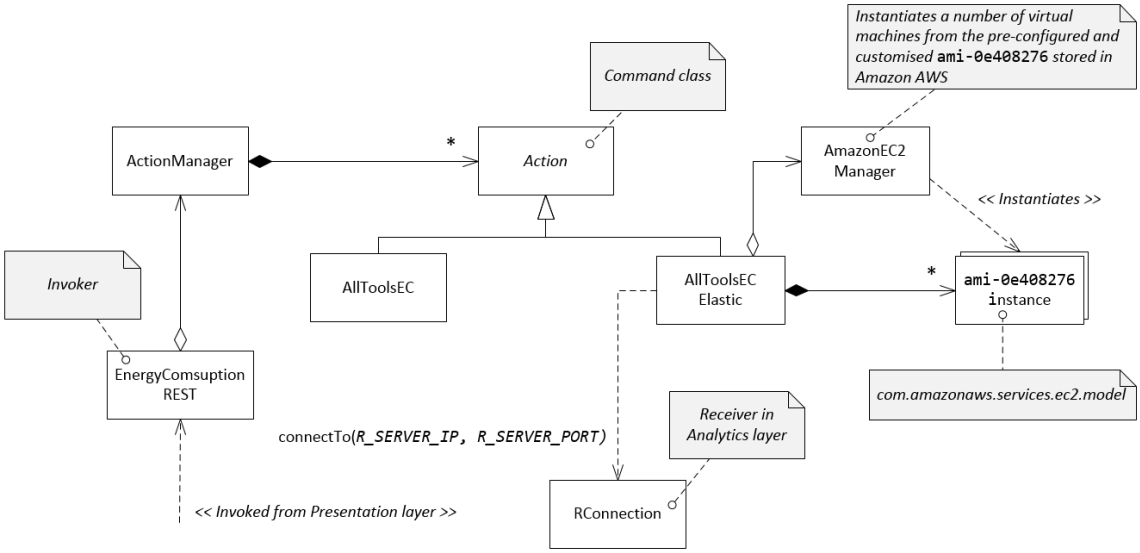


Figure 10: Class diagram of the distributed Energy Consumption use case. AllToolsECElastic interacts with AmazonEC2Manager for the allocation and disposal of AMI instances.

6.2. Cloud Deployment

A specific AMI equipped with R components, ProActive and PARConnector libraries was created for deploying the enhanced solution in the cloud. Thus, commanded by the **AllToolsECElastic** class, the **AmazonEC2Manager** will draw a fixed number of AMI instances every time distributed analytics is required. Apart from the AMI identification and instance types, other properties such as access keys, user data, security group and key name are expected to realise the dynamic provision and deprovision of computing resources. In particular, the AMIs employed here were instantiated into t2.medium types configured with two virtual CPUs and 4GB memory.

7. Discussion and Further Work

From the theoretical perspective, this work enlarged on a computing model for a data management environment that comprises shopfloor data characterisation and a topology for the systematic ingestion, transfer and access of data. To complete this, a model for a data analytics framework built on a conceptual layered architecture that separates components by functionality and exploits elastic computing was presented. The design and specification offers inter-module loose-coupling and intra-module scalability. The former is realised with the exploitation of REST APIs that support wide range of data formats while facilitating integration of technologies. The latter is offered by defining logical divisions that serve for the systematic grouping and deployment of use case

functionality. To demonstrate this, the models were implemented using standard open source processing platforms and big data technologies like Apache Storm, Kafka, CouchDB, JSON, R, Rserve and ProActive since they enable interoperability and efficiently cooperate with current machinery to automate production [42]. Although their end-to-end interplay was illustrated in the machining context, the computing elements are generic enough to handle other case studies.

From the practical perspective, the MC-Monitor manages both streamed and offline data, and its deployment is specified using CloudML which, in turn, can be automatically deployed on and adapted to different IaaS cloud providers, hence reducing costs and deployment difficulty while offering better control over software platforms employed. Moreover, it is easy (some time even semi-automatic) to add new sensors to the machine and to ingest their data for analytics purposes. The MC-Analytics supports integration of knowledge bases, analytic components and end-user visualisations. Consequently, offering scalability, effectiveness, extensibility and reusability for maintaining a competitive edge [43]. The MC-Monitor and the MC-Analytics were demonstrated over a IaaS cloud infrastructure, as this provides better control over the software platforms employed, thus allowing fine tuning for configuration, optimization and rapid deployment. Not only the modules have delivered a flexible solution baseline for realising key goals of the MC-Suite project but also, more importantly, the project itself has been instrumental in offering the right context to validate these modules. This is reflected in terms of realistic scenarios, case studies and feedback provided by industrial partners.

Initial trials of the approach were accepted by end users, however some limitations have been identified. These include the prototyping and testing of preprocessing mechanisms since they rely on a distributed and complex framework that requires integration with on-site technology, data centres, edge computing devices or legacy infrastructures. Also, cyber security penetration is left unaddressed. Therefore, the best standard approaches to address vulnerability and add resilience to cope with disturbances and respond in acceptable time should be explored. Additionally, the Energy Consumption use case verifies this end-to-end solution, however the integration of more advanced analytics as well as streamed data should be demonstrated to evaluate advantages. Although the main goal of the industrial partner was to equip operators and line managers with an analytics dashboard for decision making, the backward integration, i.e. from the analytics in the cloud to the manufacturing process remains an interesting area to seek further development.

8. Conclusions

This paper contributed with the design and application of two modules: a data management environment and a data analytics framework. The first module was designed to collect, process, transmit and store continuously generated manufacturing shopfloor data. In order to take advantage of this, the second module was created for mining high volumes of manufacturing data in

the cloud. In addition, due to the high complexity, multiple sources of information and large data sets, an elastic computing model was introduced as an extension to the data analytics framework. This combination results in a flexible, distributable and scalable approach that offers a seamless integration of technologies capable to adapt to different factory settings as well as to cloud computing providers. Therefore, reducing infrastructure costs and minimising deployment difficulty.

9. Acknowledgment

The authors would like to thank the support of the Horizon 2020 MC-SUITE (ICT Powered MaChining Suite) project funded by the European Commission under grant agreement N^{o} 680478 `http://www.mc-suite.eu`, and insightful feedback within the EPSRC Digital Manufacturing on a Shoestring *EP/R*032777/1.

References

[1] Andal-Ancion, A., Cartwright, P., Yip, G.S. 2003. Digital Transformation of Traditional Businesses. *MIT Sloan Management Review*, 44(4):34–41.

[2] Lyytinen, K., Y. Yoo and R.J. Boland Jr. 2016. Digital product innovation within four classes of innovation networks. *Info Systems J* 26:47–75. doi:10.1111/isj.12093

[3] Lasi, H., P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffman. 2014. "Industry 4.0". *Business & Information Systems Engineering* 6(4):239–242.

[4] Federal Ministry of Education and Research. 2013. "Project of the future: Industry 4.0". Available at `https://industrie40.vdma.org/en/ueber-uns`.

[5] Monostori, L. 2014. "Cyber-physical Production Systems: Roots, Expectations and R&D Challenges, Variety Management in Manufacturing". *Procedia CIRP* 17:9–13. doi:10.1016/j.procir.2014.03.115

[6] Ardito, L., A.M. Petruzzelli, U. Panniello, A.C. Garavelli. 2018. Towards Industry 4.0: Mapping digital technologies for supply chain management-marketing integration. *Business Process Management Journal*. doi:10.1108/BPMJ-04-2017-0088

[7] Ciffolilli, A. and A. Muscio. 2018. "Industry 4.0: national and regional comparative advantages in key enabling technologies". *European Planning Studies*, 26(12):2323–2343. doi:10.1080/09654313.2018.1529145

[8] Ferreira, L., G. Putnik, M. Cunha, Z. Putnik, H. Castro, C. Alves, V. Shah and M.L.R. Varela. 2013. "Cloudlet architecture for dashboard in cloud and ubiquitous manufacturing". *Procedia CIRP* 12:366–371. doi:10.1016/j.procir.2013.09.063

[9] Kiirikki, J. and M. Haag. 2013. "Ubiquitous Assembly Cell Concept and Requirements". *Procedia CIRP* 12:157–162. doi:10.1016/j.procir.2013.09.028

[10] Lee, K.C., N. Chung, and J. Byun. 2015, "Understanding continued ubiquitous decision support system usage behavior". *Telematics and Informatics* 32(4): 921–929, 2015. doi:10.1016/j.tele.2015.05.001

[11] Horváth, I. and R.W. Vroom. 2015. "Ubiquitous computer aided design: A broken promise or a Sleeping Beauty?". *Computer-Aided Design* 59:161–175. doi:10.1016/j.cad.2014.10.006

[12] Botta, A., W. de Donato, V. Persico, and A. Pescapé. 2016. "Integration of Cloud computing and Internet of Things". *Future Generation Computer Systems* 56(C): 684–700. doi:10.1016/j.future.2015.09.021.

[13] Cheng, B., L. Shu, P. Li, M. Mukherjee, and B. Yin. 2011. "Smart Factory of Industry 4.0: Key Technologies, Application Case, and Challenges". *IEEE Access* 6:6505–6519. doi:10.1109/ACCESS.2017.2783682

[14] Jagadish, H.V., J. Gehrke, A. Labrindis, Y. Papakonstantinou, J.M. Patel, R. Ramakrishnan, and C. Shahabi. 2014. "Big data and its technical challenges". Communications ACM 57(7):86–94. doi:10.1145/2611567

[15] Ardito, L., D. D'Adda, A.M. Petruzzelli. 2018. "Mapping innovation dynamics in the Internet of Things domain: Evidence from patent analysis". *Tehnological Forecasting & Social Change*, 136:317–330. doi:10.1016/j.techfore.2017.04.022

[16] Zhong, R.Y., C. Xu, C. Chen, and G.Q. Huang. 2017. "Big Data Analytics for Physical Internet-based intelligent manufacturing shop floors". *International Journal of Production Research* 55(9):2610–2621. doi:10.1080/00207543.2015.1086037

[17] Woo, J., S-J. Shin and W. Seo. 2016. "Developing a Big Data Analytics Platform for Increasing Sustainability Performance in Machining Operations". International Conference on Flexible Automation and Intelligent Manufacturing, Korea, June 27-30.

[18] Windmann, S., A. Maier, O. Niggemann, C. Frey, A. Bernardi, Y. Gu, H. Pfrommer et al. 2015. "Big Data Analysis of Manufacturing Processes". *Journal of Physics: Conference Series* 659(1):12055. doi:10.1088/1742-6596/659/1/012055

[19] Mourtzis, D., E. Vlachou, N. Xanthopoulos, M. Givehchi and L. Wang. 2016. "Cloud-based adaptive process planning considering availability and capabilities of machine tools". *Manufacturing Systems* 39:1–8. doi: 10.1016/j.jmsy.2016.01.003

[20] Wan, J., T. Shenglong, D. Li, C. Liu and H. Abbas. 2017. "A Manufacturing Big Data Solution for Active Preventive Maintenance", *IEEE Transactions on Industrial Informatics* 13(4):2039–2047. doi:10.1109/TII.2017.2670505

[21] Wang, S., J. Wan, M. Imran, D. Li and C. Zhang. 2018. "Cloud-based smart manufacturing for personalized candy packing application", *J Supercomput* 74:4339–4357. doi:10.1007/s11227-016-1879-4

[22] Chen, H., X. Fei, S. Wang, X. Lu, G. Jin, W. Li, and X. Wu. 2014. "Energy Consumption Data Based Machine Anomaly Detection". International Conference on Advanced Cloud and Big Data, China, November 20-22. doi:10.1109/CBD.2014.24.

[23] O'Donovan, P., K. Leahy, K. Bruton, and D.T.J. O'Sullivan. 2015. "An industrial big data pipeline for data-driven analytics maintenance applications in large-scale smart manufacturing facilities". *Journal of Big Data* 2(1):25. doi:10.1186/s40537-015-0034-z

[24] Park, J. and S. Chi. 2016. "An implementation of a high throughput data ingestion system for machine logs in manufacturing industry". Conference on Ubiquitous and Future Networks, Austria, July 5-8. doi:10.1109/ICUFN.2016.7536997

[25] Wang, J and J. Zhang. 2016. "Big data analytics for forecasting cycle time in semiconductor wafer fabrication system". *International Journal of Production Research* 54(23):7231–7244. doi:10.1080/00207543.2016.1174789

[26] Zhong, R.Y., S.T. Newman, Q.G. Huang, and S. Lan. 2016. "Big Data for supply chain management in the service and manufacturing sectors: Challenges, opportunities, and future perspectives". *Computers & Industrial Engineering* 101:572–591. doi:10.1016/j.cie.2016.07.013

[27] Zhang, Y., S. Ren, Y. Liu, and S. Si. 2017. "A big data analytics architecture for cleaner manufacturing and maintenance processes of complex products". *Cleaner Production* 142(2):626–641. doi:10.1016/j.jclepro.2016.07.123.

[28] Heinrichs, W. 2005. Design Management - Do it Anywhere. *Electronics Systems and Software* 3(4):30–33.

[29] Wu, D., W. Rosen, L. Wang, and D. Shaefer. 2015. "Cloud-based design and manufacturing: A new paradigm in digital manufacturing and design innovation". *Computing Aided Design* 59:1–14. doi:10.1016/j.cad.2014.07.006.

[30] SSAI Expert Group. 2010. The Future of Cloud Computing. Technical Report. Available at `http://cordis.europa.eu/fp7/ict/ssai/docs/cloud-report-final.pdf`

[31] SSAI Expert Group. 2012. A Roadmap for Advanced Cloud Technologies under H2020. Technical Report. Available at `http://cordis.europa.eu/fp7/ict/ssai/docs/cloud-expert-group/roadmap-dec2012-vfinal.pdf`

[32] Ferry, N., A. Rossini, F. Chauvel, B. Morin, and A. Solberg. 2013. "Towards model - driven provisioning, deployment, monitoring, and adaptation of multicloud systems". IEEE International Conference on Cloud Computing, USA, June 28 - July 3. doi:10.1109/CLOUD.2013.133

[33] Verl, A., E. Abele, U. Heisel, A. Dietmair, P. Eberspächer, R. Rahäuser, S. Schrems et al. 2011. "Modular Modeling of Energy Consumption for Monitoring and Control". International Conference on Life Cycle Engineering, Germany, May 2-4. doi:10.1007/978-3-642-19692-8_59

[34] Ferry, N., G. Terrazas, P. Kalweit, A. Solberg, S. Ratchev, and D. Weinelt. 2017. "Towards a Big Data Platform for Managing Machine Generated Data in the Cloud". IEEE International Conference on Industrial Informatics, Germany, July 24-26. doi:10.1109/INDIN.2017.8104782

[35] Hashem, I., I. Yaqoob, N. Anuar, S. Mokhtar, A. Gani, and S. Ullah Khan. 2017. "The rise of big data on cloud computing: Review and open research issues". *Information Systems* 47:98–115. doi:10.1016/j.is.2014.07.006

[36] Pokorny, J. 2013. "NoSQL databases: a step to database scalability in web environment". *International Journal of Web Information Systems* 9(1):69–82. doi:10.1108/17440081311316398.

[37] Richards, M. 2015. Software Architecture Patterns, O'Reilly Media, Inc.

[38] Cockburn, A. 1996. "Prioritizing Forces in Software Design", In *Pattern Languages of Program Design 2*, edited by Vlissides et al., 319–333. Addison-Wesley Longman Publishing Co., Inc.

[39] Herbst, N.R., S. Kounev, and R. Reussner. 2013. "Elasticity in Cloud Computing: What It Is, and What It Is Not". International Conference on Autonomic Computing, USA, June 26-28.

[40] Galante, G. and L.C. de Bona. 2012. "A Survey on Cloud Computing Elasticity". IEEE International Conference on Utility and Cloud Computing, USA, November 5-8. doi:10.1109/UCC.2012.30

[41] Coutinho, E.F.; F.R. de Carvalho Sousa, P.A.L. Rego, D.G. Gomes, and J.N. de Souza. 2015. "Elasticity in cloud computing: a survey". *Annals of Telecommunications* 70(7–8): 289–309. doi:10.1007/s12243-014-0450-7.

[42] Yang, C., W. Shen, and X. Wang. 2018. "The Internet of Things in Manufacturing". IEEE International Conference on Computer Supported Cooperative Work in Design, China, May 4-6.

[43] Terrazas, G., L. De Silva, and S. Ratchev. 2019. "Towards a Cloud-Based Analytics Framework for Assembly Systems". In *Precision Assembly in the Digital Age* 530:134–141, doi:10.1007/978-3-030-05931-6_13.