big data and cognitive computing

MDPI

Article

Big Data Processing and Analytics Platform Architecture for Process Industry Factories

Martin Sarnovsky *, **Peter Bednar** and **Miroslav Smatana**

Department of Cybernetics and Artificial Intelligence, Technical University Kosice, Letna 9, 04001 Kosice, Slovakia; peter.bednar@tuke.sk (P.B.); miroslav.smatana@tuke.sk (M.S.)
* Correspondence: martin.sarnovsky@tuke.sk

Received: 26 November 2017; Accepted: 23 January 2018; Published: 26 January 2018

Abstract: This paper describes the architecture of a cross-sectorial Big Data platform for the process industry domain. The main objective was to design a scalable analytical platform that will support the collection, storage and processing of data from multiple industry domains. Such a platform should be able to connect to the existing environment in the plant and use the data gathered to build predictive functions to optimize the production processes. The analytical platform will contain a development environment with which to build these functions, and a simulation environment to evaluate the models. The platform will be shared among multiple sites from different industry sectors. Cross-sectorial sharing will enable the transfer of knowledge across different domains. During the development, we adopted a user-centered approach to gather requirements from different stakeholders which were used to design architectural models from different viewpoints, from contextual to deployment. The deployed architecture was tested in two process industry domains, one from the aluminium production and the other from the plastic molding industry.

Keywords: Big Data analytics; Big Data architecture; process industries; predictive data analysis

1. Introduction

Process industries represent a significant share of the European industry in terms of energy consumption and environmental impact. In this area, optimization can lead to significant savings, both economic and environmental. Predictive modeling can prove effective when applied to the optimization of production processes. However, the application of these techniques is not straightforward. Predictive models are built using the data obtained from production processes. In many cases, process industries must invest in the monitoring and data integration as well as in the development and maintenance of the underlying infrastructure for data analytics. Many other obstacles are also present, e.g., interoperability issues between software systems in production, difficulties in the physical monitoring of the production parameters, problems with the real-time handling of the data, or difficulties in defining relevant Key-Performance Indicators (KPIs) to support management. Therefore, the deployment of such predictive functions in production with reasonable costs requires consolidation of the available resources into shared cloud-based technologies. In the case of more flexible production environments, even more significant approaches are possible, such as the reinvention or redesign of the production processes. However, this is not applicable to major, capital intensive process industries. In this case, the integration of innovations in the established production processes can be fundamental in their transformation from resource-consuming production into the “circular” model.

The work presented in this paper was performed as part of the MONSOON (MOdel based control framework for Site-wide OptimizatioN of data-intensive processes) project (https://www.spire2030.eu/monsoon), focused on sectors of the aluminium and plastic industries. The main objective was to pursue process optimization from the perspective of raw materials and energy reduction, through the application of optimizations distributed across multiple production units within multiple distributed production sites.

This paper aims to introduce a cross-sectorial, scalable, Big Data analytics platform for such process industries, which is capable of storing and processing large amounts of data from multiple sites. Besides cost reduction, sharing the platform for multiple sites from different sectors will enable the transfer of the best practices and knowledge across different domains. The paper is organized as follows: Section 2 gives an overview of the related research and projects. Section 3 gives an outline of the design methodology and specification of the user requirements. Sections 4–6 present the architectural views and describe the main components of the platform from different viewpoints.

2. Related Work

In the process industries, production processes must be accurately modelled to achieve optimization goals set by the companies. Modeling using predictive models allows companies to adapt to the changing requirements as the process models are implemented in an environment that provides further knowledge through simulation and optimization. The traditional approach is based on mathematical modeling complemented with statistical modeling or data-driven empirical methods [1–3]. To reduce computational effort, various methods can be used [4–6]. Process analytical technologies are widely used, and data analysis plays an important role in any modern process industry, mainly in analyzing and controlling manufacturing processes and attributes of the materials used. In this case, real-time optimization can be an effective approach to the improvement of processes and operation. A model predictive control strategy, machine learning techniques, self-optimizing and control mechanisms [7–10] are useful tools in the construction, adaptation and application of real-time optimization methods applied in the process industries. Several projects are also aimed at using machine learning and predictive functions in the process industries. FUDIPO [11] integrates the machine learning functions into the process industries to provide improvements in energy and resource efficiency. In OPTICO [12], a partial least squares model is used to predict the mean particle size of polystyrene beads based on the collected data. Another example is ProPAT [13], where production optimization is achieved by extracting information and knowledge using data analytic tools. ProPAT also aims to develop novel and smart sensors for measuring process parameters, and integrate them into a global control platform for data acquisition, processing and mining. The platform provides self-learning and predictive capabilities aimed at reducing over-costs derived from the deviations from the optimum process. CONSENS [14] introduces online sensing equipment and control of the key product parameters in the production of high-value products with high-quality demands. The COCOP project [15] will enable plant-wide monitoring and control by using the model-based, predictive, coordinating optimization concept with integration of the plant's automation systems.

From the perspective of existing platforms and technologies capable of handling of big volumes of data, numerous technologies are present that address various aspects of Big Data processing [16]. In areas of data ingestion, several technologies exist including Sqoop, Flume or Kafka. MQTT (MQ Telemetry Transport or Message Queuing Telemetry Transport) is a widely used, simple and lightweight messaging protocol [17]. The processing frameworks have been grouped according to the state of the data they are designed to handle. Some systems handle data in batches, while others process data in a continuous stream as it flows into the system. Batch processing involves operating over a large, static dataset and returning the result later when the computation is complete. Most popular is Hadoop, based on MapReduce—a programming model that allows the processing of Big Data in a parallel and distributed way in a cluster. Stream processing compute over data as it enters the system. This requires a different processing model than the batch paradigm. Current architectures of Big Data processing platforms require technologies that can handle both batch and stream workloads [18]. These frameworks simplify diverse processing requirements by allowing the same or related components and application programming interfaces (APIs) to be used for both types of data. Apache Spark [19] is a next generation batch processing framework with stream processing capabilities. Built using many of the same principles as Hadoop's MapReduce engine, Spark focuses primarily on speeding up batch processing workloads by offering full in-memory computation and processing optimization. Apache Flink is an open source framework for distributed stream processing. Flink uses the concepts of streams for all applications. In Flink's terms, a batch is a finite set of streamed data. Regarding streaming, Flink processes data streams as true streams, i.e., data elements are immediately pipelined through a streaming program as soon as they arrive, which distinguishes Flink from Spark (which performs microstreaming). To enable machine learning from Big Data, several tools are available. Spark MLlib is a machine learning library of Spark. The library comes with machine learning algorithms for classification, regression, clustering, and collaborative filtering. FlinkML is the machine learning library for Apache Flink. FlinkML has a scikit-inspired pipeline mechanism which allows the data scientists to quickly build complex data analysis pipelines the way they appear in the daily works of data scientists. H_2O (0xdata, Mountain View, USA) is another open source software for Big Data analysis. The machine learning library scales statistics, machine learning and mathematics over Big Data. Currently popular frameworks are TensorFlow and Keras. TensorFlow is a symbolic math library used for a wide array of machine learning and other applications, which is platform agnostic and supports both CPUs and (multiple) GPUs. Keras is a machine learning framework for Python built on top of abstractions of other back-end libraries (e.g., TensorFlow, Theano, Deeplearning4j, CNTK, MXNET).

3. Requirements Engineering and Design Methodology

This section gives an overview of the requirements engineering approach used in the design of the analytical platform. We adopted the user-centered design [20] approach. We chose that approach instead of specifying the requirements at the beginning for various reasons. An incomplete requirements analysis performed at the beginning often leads to later problems in the system development. ISO 9241-210 [21] gives guidance on human-centered design activities throughout the whole life cycle of computer-based interactive systems. A user-centered design aims to establish communication between users and developers and to gather different requirements on each side. In [22], the authors show that iterative approaches drastically reduce the gap between users and developers and their understanding of the developed system.

In the process of architecture design, we followed the standards and best practices in that area and were guided by ISO/IEC/IEEE 42010:2011 "Systems and software engineering—Architecture description" [23], which establishes a methodology for the architecture description of software systems. It involves several steps which include specifying the architectural viewpoints that address stakeholders' concerns and creating consistent architectural views (for each viewpoint) with architectural models. As defined in [24], architecture comprises concepts or properties of a system in its environment, embodied in its elements, relationships, and in the principles of its design and evolution.

In the architecture design, we employed the approach described in [24]. The stakeholders were involved in formulating their needs during the development of scenarios and subsequent requirements for the engineering process. The process started with the main vision scenario, and more specific context scenarios were derived from the main one. The context scenarios aimed to capture the specific context, e.g., use for a certain user role, and were used to illustrate the benefits of the platform for specific users in their specific tasks [25]. Another source of requirements were workshops and interviews conducted with the industrial partners in the project in order to gain insights into the roles, processes or problems in application domains. The selected interview partners (project partners involved in the operation of the technology in both domains) covered different areas of the working processes that are relevant to the MONSOON project. The main objective was to get a better understanding of the work performed in each domain, which would provide better context for deriving requirements to architecture. The interviews started with an introduction to the MONSOON project goals, and then different areas of relevant production processes were discussed. The documentation of this information was done by taking direct notes or by capturing the environment by taking pictures.

Those data were gathered and analyzed, and led to the identification of the first set of requirements. Based on those requirements, we created a draft of the architectural description based on which the first prototype was created. We decided on the following viewpoints from which the architectural views were derived:

- *Context viewpoint*—describes a broader context of the system—relationships and dependencies with other systems and environment
- *Information viewpoint*—describes data models and the data flow and how the data are manipulated and stored
- *Functional viewpoint*—describes the main functional elements of the architecture and interfaces and interactions
- *Deployment viewpoint*—describes how and where the system is deployed, considering hardware and physical dependencies.

4. Context View

The context viewpoint describes interactions, relationships and dependencies between the system and its environment which interact with the system, such as other systems, users, or developers.

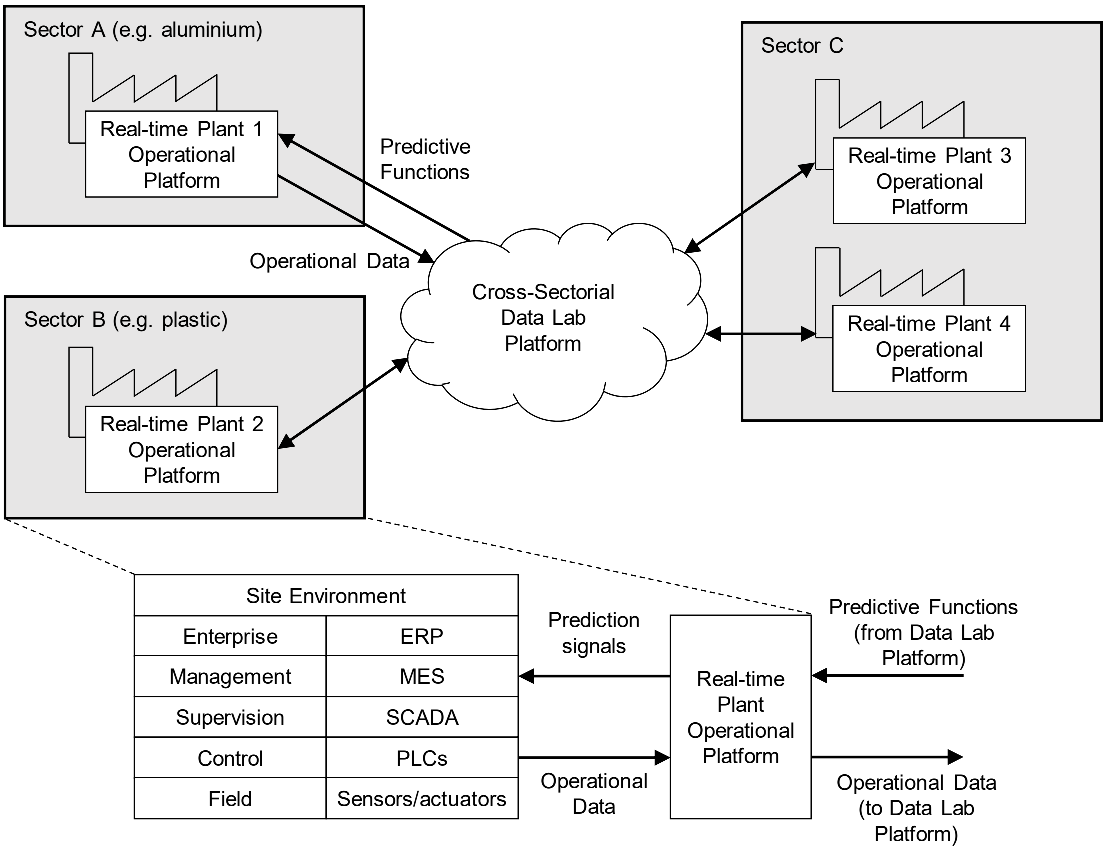
The proposed architecture was designed in the context of the MONSOON, a SPIRE (Sustainable Process Industry through Resource and Energy Efficiency) research project that aims to develop an infrastructure in support of the process industries. Its main objective is to establish the data-driven methodology which will support the identification and exploitation of optimization potentials by applying model-based predictive controls in the production processes.

To validate and demonstrate the results, two real environments are used within the project: an aluminium plant in France and a plastic factory in Portugal. We have identified two main use cases for both domains. For the aluminium sector, we focused on production of the anodes (positive electrodes) used in aluminium extraction by electrolysis. The first use case was targeted to predictive maintenance, where the main objective was to anticipate the breakdowns and/or highlight equipment/process deviations that impact the green anode final quality (e.g., anode density). The second case dealt with the predictive anode quality control, where the goal was to identify bad anodes with a high level of confidence and scrap them to avoid sending them to the electrolysis area.

For the plastic domain, the first case was from the area of production of coffee capsules, produced in large quantities with little variation and relatively low quality specifications. In this type of production, it is important to produce the correct diameter and height of the coffee capsules and to make sure that the holes at the bottom of the capsules are formed properly. The second use case covered the production of the parts used in the automotive industry, where methods of over-molding metal inserts are applied. Based on the identified use cases, we divided the data analytics architecture of two main components as shown in Figure 1:

- *Real Time Plant Operation Platform*—used during runtime, can be used by employees working on the shop floor. The component communicates with the existing heterogeneous systems deployed on the production site including sensors or systems such as ERP (Enterprise resource planning), SCADA (Supervisory control and data aquisition), MES (Manufacturing execution system) and others. Relevant data from the production site are transferred to the Cross-Sectorial Data Lab.
- *Cross-Sectorial Data Lab*—a collaborative environment where high amounts of data from multiple sites are collected and processed in a scalable way. Designed to be used by the data scientists or the global process manager. It consists of Big Data storage and processing elements. It also contains development tools for the creation of predictive functions, simulation tools for evaluation of those functions in the testing environment and their deployment in the production site, and a semantic framework which provides a common language between the data scientists and domain experts (e.g., process managers).

The components are connected by the specified interfaces. The components and corresponding interfaces are described in more detail in the Functional View section.



Sector A (e.g. aluminium)
Real-time Plant 1 Operational Platform
Predictive Functions
Operational Data
Sector B (e.g. plastic)
Real-time Plant 2 Operational Platform
Cross-Sectorial Data Lab Platform
Sector C
Real-time Plant 3 Operational Platform
Real-time Plant 4 Operational Platform
Prediction signals
Operational Data
Real-time Plant Operational Platform
Predictive Functions (from Data Lab Platform)
Operational Data (to Data Lab Platform)

Site Environment	
Enterprise	ERP
Management	MES
Supervision	SCADA
Control	PLCs
Field	Sensors/actuators

Figure 1. High level architecture overview. Abbreviations: ERP, Enterprise resource planning; SCADA, Supervisory control and data acquisition; MES, Manufacturing execution system; PLC, Programmable Logic Controller.

5. Information View

The Information View is formalized in the form of the process models and semantic meta-data models organized in the Semantic Modelling Framework. The main goal of the semantic modeling is to provide a common communication language between domain experts and stakeholders and data scientists. On the one hand, data scientists need a deep knowledge of the business objectives and modelled phenomena acquired from the stakeholders and domain experts. On the other hand, stakeholders and domain experts need to interpret the results of the data analysis.

The Semantic Modelling Framework combines concepts from the IEC/ISO standards for the enterprise-control system integration with concepts from the data mining methodologies and interoperability standards such as Cross-Industry Standard Process for Data Mining (CRISP-DM) [26,27], Predictive Model Markup Language (PMML) [28], and Portable Format for Analytics (PFA) [29].

The overall structure of the Semantic Modelling Framework is presented in Figure 2 and covers the following main concepts: Production Processes and Process Segments, Equipment and Physical Assets, Key-Performance Indicators, Data Elements and Predictive Functions.

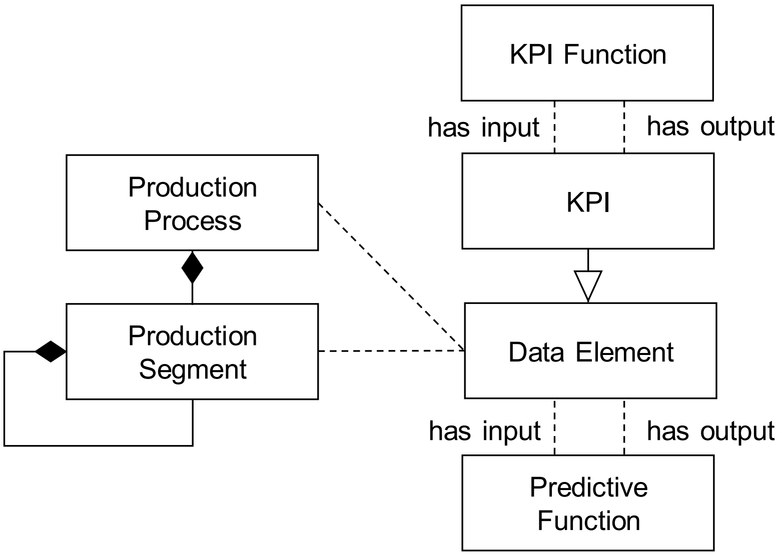


Figure 2. The main concepts of the Semantic Modelling Framework. Abbreviation: KPI, Key-Performance Indicator.

5.1. Production Processes and Process Segments

Production Process and Process Segments concepts represent the decomposition of the overall production process into production steps. The concept of Production Segment represents a logical grouping of the related Key-Performance Indicators, Data Elements and resources (i.e., Equipment and Physical Assets) required to carry out a production step. The model of the Production Process is specified using the workflow graphical notation.

5.2. Equipment and Physical Assets

Equipment and Physical Assets concepts describe sites, areas, production units, production lines, work cells, process cells, or units linked to the specific Process segment. Equipment may be hierarchically made up of several other sub-pieces of equipment (for example, a production line may be made up of work cells etc.). Each may be defined as a separate equipment element with separate properties and capabilities. Additionally, a grouping of equipment with similar characteristics and purposes can be described by Equipment Classes. Any piece of equipment may be a member of zero or more equipment classes.

5.3. Key-Performance Indicators

The concepts of KPIs represent metrics designed to visualize, assess and manage the performance or impact of specific operations within the process industries. They are linked to the Process Segments, or specified for the overall Production Process. The main properties of the KPI metrics (e.g., type and quantity) are inherited from the Logical Data Elements (see the description in the following subsection). The KPIs can be grouped into KPI categories according to the classification in ISO 22400 [30].

Besides this classification, causal relations between KPIs can be expressed by the KPI Functions, which transform one KPI to another (e.g., material or energy savings can be converted into KPIs for the environmental impact, such as level of emissions). The input of the KPI function can be also Evaluation Metrics, which estimate the performance of the predictive functions (see the description of the Predictive Functions in the subsequent subsection). Using the composition of KPI Functions, it is possible to infer the impact of the deployment of a predictive function on the performance of the production process, where the performance can be evaluated from various perspectives such as material/energy consumption, product quality, environmental impact, etc.

5.4. Data Elements

Data elements are modelled in two levels of abstraction: *logical* and *physical*. A Logical Data Element concept describes the role, type and quantity (e.g., units of measurements) of any data element related to the production step or equipment. A data element can have an input role (for measurements or input control signals) or output role (for diagnostic signals). Output data elements can be selected as a target attribute for the predictive function. One data element can have both roles depending on the goal of the data analysis (i.e., a control signal can be an input for the predictive maintenance or output for the predictive control). The type of the data elements characterizes elements according to their values and denotes continuous, ordinal, nominal, spatial or series data elements.

Physical Data Elements link Logical Data Elements to the physical representation of the data in the structured records or files. Multiple Physical Data Elements (fields) can be grouped into one record or file and described by a Physical Schema. Each Physical Data Element has a specified value type. The value type can be primitive (e.g., byte, integer, string) or complex (array, enumeration, map or union of types).

5.5. Predictive Functions

Similarly to Data Elements, Predictive Functions are modelled at the physical and logical levels. The concept of Predictive Functions specifies which data elements are the inputs to the predictive function (i.e., predictors or independent variables) and which data elements are the outputs (i.e., dependent or predicted variables). Additionally, Physical Predictive Functions specify details about the training data and algorithm used for building a specific predictive function and Evaluation Statistics, describing the performance of the predictive function as evaluated on the validation set or real-time operational data.

6. Functional View

There are two main components of the architecture: the Data Lab as the platform for data storage and processing and the Plant Operation Platform deployed on-site and providing a connection to the production environment.

The Data Lab platform is connected to the Plant Operation Platform using the following interfaces:

- *Data Replication Service*—This interface allows the uploading of a large batch of historical data collected on the site to the cloud Data Lab storage.
- *Messaging Service*—This interface allows real-time asynchronous communication between the Data Lab and Operational platform. It is optimized for frequently updated data (e.g., from sensors) with a relatively small size of the payload for each update.
- *Functions Repository*—This interface allows the export of predictive functions built on the Data Lab platform and their deployment on the Plant Operation Platform for scoring of the operational data.

All interfaces are provided by the Data Lab platform with Plant Operation Platform components acting as clients.

6.1. Plant Operational Platform

The architecture of the Plant Operation Platform is presented in Figure 3. The main component is the Virtual Process Industries Resources Adapter, which integrates the Run-time Container and the Operation Data Visualization Framework, and streams integrated data to the Data Lab platform using the Data Replication Service and Messaging Service.

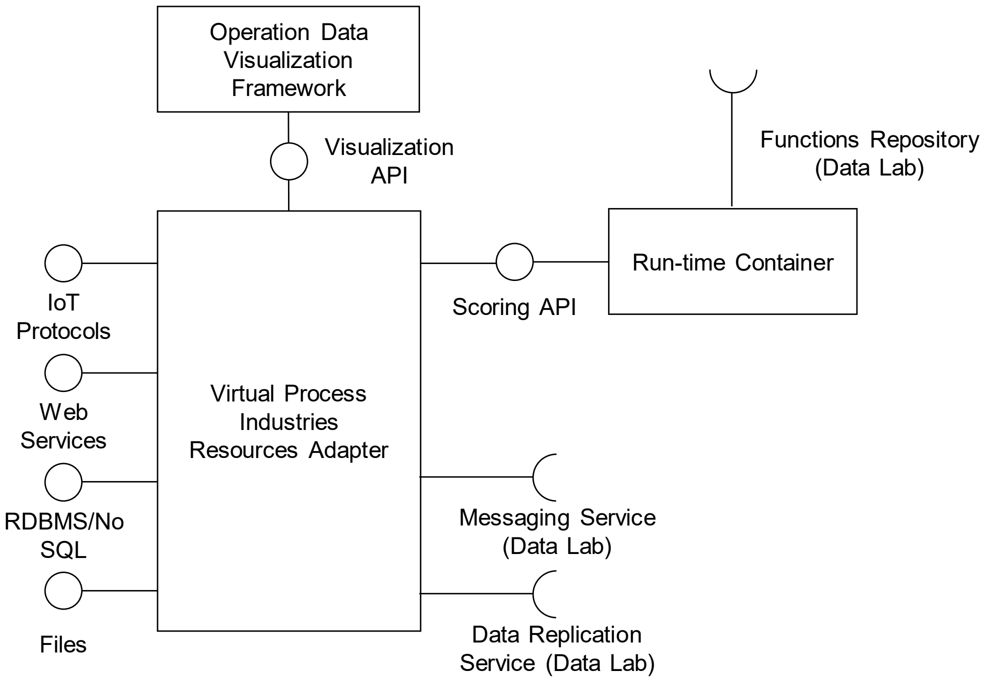

Operation Data Visualization Framework
Visualization API
Functions Repository (Data Lab)
IoT Protocols
Web Services
RDBMS/No SQL
Files
Virtual Process Industries Resources Adapter
Scoring API
Run-time Container
Messaging Service (Data Lab)
Data Replication Service (Data Lab)

Figure 3. Plant Operational Platform architecture. Abbreviations: IoT, Internet of Things; RDBMS, relational database management system; NoSQL, Non Structured Query Language; API, application programming interface.

6.1.1. Virtual Process Industries Resources Adapter

The main function of the Virtual Process Industries Resources Adapter (VPIRA) is data integration, mediation and routing. VPIRA provides the main integration layer between the cloud-based Big Data processing and analytical platform, and the local plant environment.

VPIRA has modular internal architecture, which consists of connector components for various data sources and targets such as IoT protocols, web services (XML SOAP or REST—Representational state transfer), relational databases (through ODBC/JDBC standards), NoSQL databases and local/remote file systems. The connectors allow the integration of data from various SCADA, MES and ERP systems deployed on the plant site. The integrated data are internally routed by the data flow engine. The data flows can be reconfigured in a flexible way, connecting multiple sources to the multiple targets, overcoming any data heterogeneity problems. Routing of data from the source to target connectors can be dynamic depending on the type of data or actual content.

Besides the flexible configuration interface, the Virtual Process Industries Resources Adapter should provide a flexible programming interface to simply implement connectors or processors for new types of data sources and formats.

6.1.2. Run-Time Container

The Run-time Container allows the import of predictive functions from the Data Lab platform and the use of these functions locally on the plant site for the real-time scoring of operational data. The predictive functions are exported from the Data Lab in the platform-independent format (see the description of the Functions Repository), which is interpreted by the internal scoring engine. The format of the predictive functions also allows interoperability with various data analytics tools not supported by the Data Lab platform. Besides scoring, the engine also performs all operations required for the pre-processing of raw data into inputs for the specific predictive function and into process prediction output. Both batch and online scoring should be supported (i.e., scoring of one or multiple records).

6.1.3. Operational Data Visualization Framework

The Operation Data Visualization Framework provides a web user interface where operational managers can configure various real-time visualizations of operational data and monitor the deployed predictive functions. The visualized data are integrated by the Virtual Process Industries Resource Adapter and can include operational data from the plant environment or predictions from the predictive functions executed in the Run-time Container. The Virtual Industries Resource Adapter streams the visualized data asynchronously through the REST web service interface provided by the Operational Data Visualization Framework. The visualized data can be enhanced with the new trend indicators using various trend analysis methods. The Operational Data Visualization Framework also provides API for the implementation of the new trend indicators.

6.2. Cross-Sectorial Data Lab Platform

The architecture of the Cross-Sectorial Data Lab platform is shown in Figure 4 and consists of the following main components: Big Data Storage and Analytics Platform; Development Tools; Semantic Modelling Tools and Simulation and Resource Optimization Framework. These components are described in more detail in the following subsections.

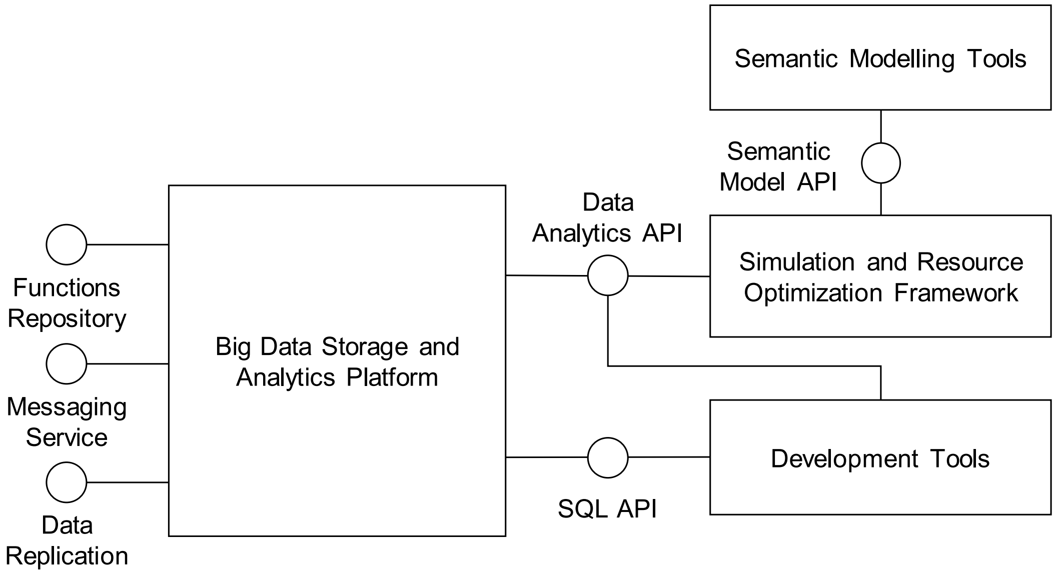


Figure 4. The architecture of the Cross-Sectorial Data Lab platform.

6.2.1. Big Data Storage and Analytics Platform

The Big Data Storage and Analytics Platform provides resources and functionalities for storage as well as for batch and real-time processing of the Big Data. It provides main integration interfaces between the site Operational Platform and the cloud Data Lab platform and the programming interfaces for the implementation of the data mining processes. The internal structure of the Big Data Storage and Analytics Platform is given in Figure 5.

Data are primarily stored in the Distributed File System, which is responsible for the distribution and replication of large datasets across the multiple servers (data nodes). A unified access to the structured data is provided by the Distributed Database using the standard SQL (Standard Query Language) interface. The main component responsible for data processing is the Distributed Data Processing Framework, which provides high-level API for the implementation of the data pre-processing tasks and for the building and validation of the predictive functions. Predictive functions are stored in the Functions Repository, where they are available for production deployment or for the simulations and overall optimization of the production processes. The rest of the components (Messaging Service and Data Replication Service) provide data communication interfaces, and connect the Operational platform to the Data Lab platform.

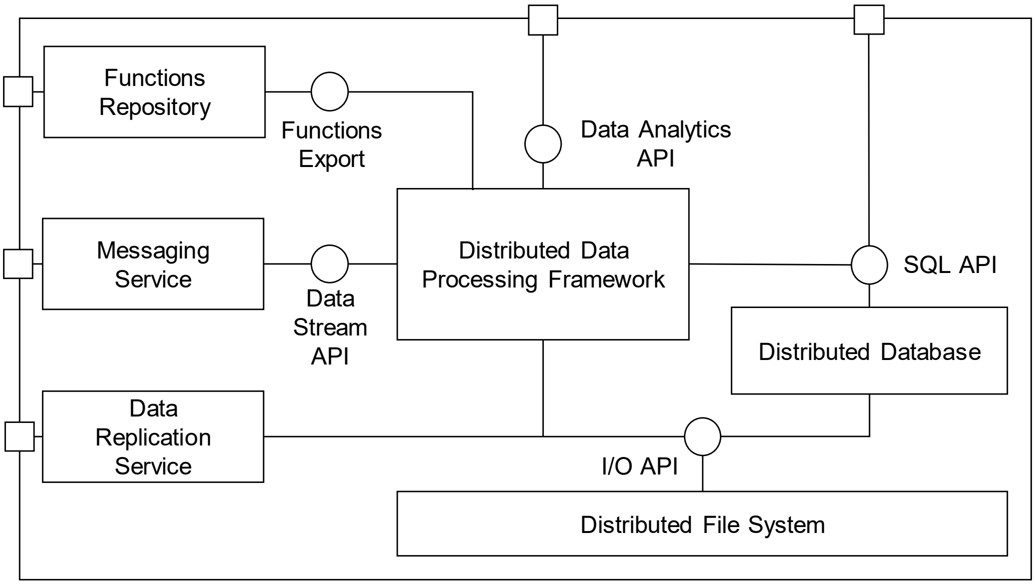


Figure 5. The internal architecture of the Big Data Storage and Analytics Platform.

The Big Data Storage and Analytics Platform consist of the following sub-components:

- *Distributed File System*—provides a reliable, scalable file system with similar interfaces and semantics to access data as local file systems.
- *Distributed Database*—provides a structured view of the data stored in the Data Lab platform using the standard SQL language, and supports standard RDBMS programming interfaces such as JDBC for Java or ODBC for Net platforms.
- *Distributed Data Processing Framework*—allows the execution of applications in multiple nodes in order to retrieve, classify or transform the arriving data. The framework provides Data Analytics APIs for two main paradigms for processing large datasets: API for parallel computation and API for distributed computation.
- *Functions Repository*—provides storage for predictive functions together with all settings required for the deployment of functions.
- *Messaging Service*—implements an interface for real-time communication between the Data Lab and Operation platforms. It provides a publish-subscribe messaging system for asynchronous real-time two-way communication, which allows to the decoupling of data providers and consumers.
- *Data Replication Service*—provides an interface for the uploading of the historical batch data between the Data Lab and Operation platform.

6.2.2. Development Tools

The Development Tools provide the main collaborative and interactive interface for data engineers, data analysts and data scientists to execute and interact with the data processing workflows running on the Data Lab platform. Using the provided interface, data scientists can organize, execute and share data, and code and visualize results without referring to the internal details of the underlying Data Lab cluster. The interface is integrated into the form of analytical ”notebooks” where different parts of the analysis are logically grouped and presented in one document. These notebooks consist of code editors for data processing scripts and SQL queries, and interactive tabular or graphical presentations of the processed data.

6.2.3. Semantic Modelling Tools

The Semantic Modelling Tools provide a collaborative user interface for the creation and sharing of semantic models specified in the Semantic Modelling Framework (see the description in the section Information View). Additionally, the Semantic Modelling Tools provide a web service interface for the storing and querying of semantic models in the machine-readable form using semantic interoperability standards such as JavaScript Object Notation (JSON) for Linked Data (JSON-LD). The web service interface allows the use of the knowledge expressed in semantic models for the optimization of the production processes in the Simulation and Resource Optimization Framework.

6.2.4. Simulation and Resource Optimization Framework

The main goal of the Simulation and Resource Optimization Framework is to support the validation and deployment of the predictive function in order to optimize overall KPIs defined for the production process. The validation is based on (a) the estimation of accuracy statistics of a predictive function (e.g., receiver operating characteristic (ROC) curves, confidence tables, etc.) on the specified validation set of historical data or real-time data stream of operational data, and (b) the computation of changes for KPIs from the accuracy statistics defined for the overall production process. This estimation of overall impacts can be used to test various "what if" scenarios, or for the automatic discrete optimization of the production process by finding the optimal combination of predictive functions for various process phases.

The Simulation and Resource Optimization Framework leverages semantic models provided by the Semantic Modelling Tools including information about the production process, KPIs and relations between KPIs and predictive function performance. The validation scripts are implemented using the Data Analytics API, and validation tasks are executed in the Distributed Data Processing Framework. Predictive functions are instantiated from the Functions repository.

Besides performing the predictive function, the Simulation and Resource Optimization Framework uses information about the "costs" associated with the input attributes, which reflect how difficult it is to obtain particular data (i.e., measure, integrate etc.). Input costs and availability in the specified production environment put additional constraints on the process optimization.

7. Deployment View

Nowadays, a large number of various technologies for Big Data processing are emerging, often with overlapping functionalities and non-functional properties. For the system architects, the selection of implementation technology is a challenging task which requires the consideration of many implementation details and compatibility constrains. The Deployment View provides consistent mapping across the existing and emerging technologies, and functional components specified in the Function View. Mapping reference implementation of the proposed architecture is summarized in Table 1 for both architecture platforms (Plant Operation platform and Cross-Sectorial Data Lab platform). All referenced technologies are available under an open source license without any usage restrictions. Alternatively, the proposed technological stack can be implemented using technologies from the main Big Data platform providers such as Cloudera, Hortonworks, IBM or Microsoft HD Insight.

Table 1. Component mapping to technologies.

Plant Operational Platform	
Virtual Process Industries Resources Adapter	Apache Nifi
Run-time Container	Hadrian reference implementation for the Portable Format for Analytics wrapped with web service interface
Operational Data Visualization Framework	Grafana
Cross-Sectorial Data Lab Platform	
Distributed File System	Apache Hadoop—HDFS
Distributed Database	Apache Spark—SQL
Distributed Data Processing Framework	Apache Spark—distributed computation DL4J (Deep Learning 4 Java) + Keras—parallel computation
Messaging Service	Apache Kafka
Data Replication Service	native WebHDFS interface proxied by Apache Knox security gateway
Development Tools	Apache Zeppelin
Common infrastructure	
Management, monitoring, provisioning configuration	Apache Ambari

In this section, we present the initial version of the platform. Our reference implementation used Apache Nifi (scalable framework for data routing, transformation) as the main technology for data integration in the Plant Operation Platform. Nifi was used for the collection of heterogeneous data from various process industry sites and to store the processed data in the Data Lab platform. The Run-time Container was based on the Hadrian scoring engine for the Portable Format for Analytics (PFA, an emerging interchange format for data-mining models). Operational data were visualized using the Grafana (https://grafana.com/), a visualization framework for analytics and monitoring. The core components of the Cross-Sectorial Data Lab platform were based on the standard Apache Hadoop (hadoop.apache.org) infrastructure, which consists of the distributed file system (Hadoop Distributed File System, HDFS) and resource manager (Yet Another Resource Negotiator, YARN). As the main framework for the implementation of the distributed data processing, Apache Spark (https://spark.apache.org) was adopted, which supports both batch and stream processing. Apache Spark also provided support for SQL data analytics. Support for parallel GPU/CPU computation was based on the combination of DL4J (https://deeplearning4j.org), an open-source, distributed deep-learning library, and Keras, a high-level neural networks API (https://keras.io). The platforms were integrated using the Apache Kafka messaging system for real-time communication and native HDFS web service interface for batch data updates. Access to the Cross-Sectorial Data Lab platform was secured by Apache Knox (https://knox.apache.org) security gateway. Cluster was managed using Apache Ambari (https://ambari.apache.org) management software and, to support the development of the predictive functions, we adopted Apache Zeppelin (https://zeppelin.apache.org/) as a basis for the Development tools component.

Figure 6 presents the Deployment View with the main types of the nodes. In order to simplify the on-site deployment, all components of the Plant Operation Platform were installed in one *Site container* server connected to the site infrastructure. This container can be deployed in the cluster for scalability and reliability. The Cross-Sectorial Data Lab platform was deployed as the cloud cluster, which consisted of three types of nodes. The security Gateway was the only server connected to the Internet, and isolated Data Lab components, which were interconnected by the private network. The *Master* ran the main services for the management, monitoring, configuration and provisioning of the Data Lab cluster such as the Apache Ambari server, HDFS Name Node and YARN resource manager. Data were stored and processed on *Worker nodes*, which ran execution services such as HDFS Data Nodes and Spark workers.

The presented, initial version of the platform was tested and evaluated in both considered domains. The main objective was to deploy the Plant operation platforms on-site to collect and transfer the data to the Cross-sectorial Data Lab. The main goal of the evaluation was to test the infrastructure for the transfer of both types of data—transfer of batches of data and real-time data streaming. The main idea was to provide a proof-of-concept, of how the platform would perform when handling both types of data obtained from real on-site plant operations.

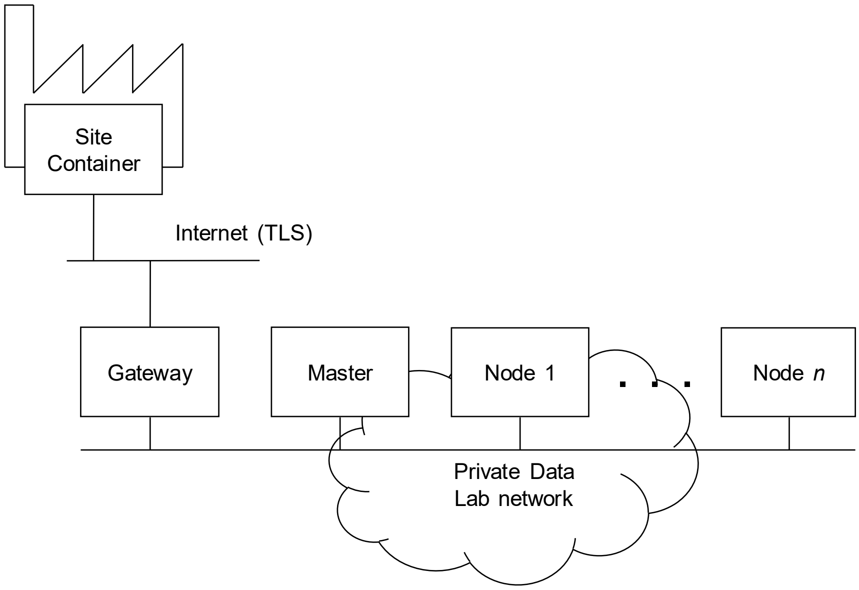


Figure 6. Deployment view with the main types of nodes.

Data were integrated using Apache Nifi and transferred using the MQTT (MQ Telemetry Transport or Message Queuing Telemetry Transport) (http://mqtt.org) protocol to Data Lab, where they were stored in an HDFS filesystem. In the aluminium domain, Historian Pi, a real-time data historian application, was used on-site and the Pi connector created one file per indicator (sensor) every 15 min (the scheduled period is configurable) on a dedicated folder in the local filesystem. Newly created files were then being sent using the Data Replication Service to the Data Lab storage component. In this case, the platform transferred and integrated approximately 10,000 sensor values from the production site every 15 min. Data were collected and replicated in the storage component. In the plastic domain, the data files were streamed into the Data Lab via Messaging service as soon as they were created, received by a subscriber, and converted to the uniform format. Then the data were stored or pushed to the visualization component for real-time analysis. For the real-time data, the platform was used to stream more than 100 sensor values per second to the storage component.

The initial version of the platform was successfully deployed and put into the operation. Data were being ingested and collected in the filesystem. The next steps involve the expansion of the platform with Development Tools, which will provide the data scientists with the environment to model, define, create, and evaluate the predictive models in the Data Lab, on top of the gathered historical data. Validated models could be then packaged and deployed to the Plant Operation Platform into the production.

8. Conclusions

In this paper, we described the complete specification of the architecture for Big Data processing in the process industries. A user-centered approach to gathering user scenarios and requirements from different stakeholders was adopted for the specification of the architecture. Based on the user scenarios and requirements, we specified the information and functional view of the architecture. The Information

View was formalized as a semantic model, which defined concepts for the modeling of the production processes, data elements, predictive functions, and inference of the Key-Performance Indicators. The Function View specifies the decomposition of architecture into modules and components, and provides a detailed functional description for each specified component and interface. From the functional point of view, the main design concept is the division of architecture into two platforms: the Plant Operation platform, deployed on-site and providing connection to the production environment, and the cloud Cross-Sectorial Data Lab platform for data storage and processing. The concept of the Data Lab platform reduces costs associated with the implementation of the data analytics methods in the production processes by consolidating the resources for multiple sites, and enables the sharing and transfer of knowledge between different industry sectors. The description of the architecture was completed with the Deployment View, which maps functional components of the technologies for Big Data processing. The main objective of the reference implementation was to provide consistent mapping of technologies with minimal dependencies, covering all functional requirements. The reference architecture was implemented and tested on two domains: the production of aluminium and the plastic molding industry. In the future, we will further focus on the standardization of the semantic models and programming application interfaces.

Acknowledgments: The work presented in the paper is supported by H2020-SPIRE-2016 Project MONSOON "MOdel based control framework for Site-wide OptimizatioN of data-intensive processes," funded by the EC within the Contract No. 723650 and by the Slovak VEGA research grant 1/0493/16 and Slovak APVV research grant APVV-16-0213.

Author Contributions: P. Bednar designed the architecture of the platform; Peter Bednar, Miroslav Smatana and Martin Sarnovsky implemented and deployed the initial version of the platform; Martin Sarnovsky and Peter Bednar wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Xie, X.; Shi, H. Dynamic Multimode Process Modeling and Monitoring Using Adaptive Gaussian Mixture Models. *Ind. Eng. Chem. Res.* **2012**, *51*, 5497–5505. [CrossRef]
2. Matzopoulos, M. Dynamic Process Modeling: Combining Models and Experimental Data to Solve Industrial Problems. In *Process Systems Engineering*; Wiley-VCH Verlag GmbH & Co. KGaA: Weinheim, Germany, 2011; pp. 1–33, ISBN 9783527631339.
3. Cameron, I.; Gani, R.; Cameron, I.; Gani, R. Chapter 6—Steady-State Process Modelling. In *Product and Process Modelling*; Elsevier: Amsterdam, The Netherlands, 2011; pp. 125–156, ISBN 9780444531612.
4. Kondov, I.; Sutmann, G. *Multiscale Modelling Methods for Applications in Materials Science*; IAS Series; Forschungszentrum Jülich, Zentralbibliothek: Jülich, Germany, 2013; ISBN 978-3-89336-899-0.
5. Weinan, E. *Principles of Multiscale Modeling*; Cambridge University Press: Cambridge, UK, 2011; ISBN 9781107096547.
6. Sadowski, T.; Trovalusci, P. (Eds.) *Multiscale Modeling of Complex Materials*; CISM International Centre for Mechanical Sciences; Springer: Vienna, Austria, 2014; Volume 556, ISBN 978-3-7091-1811-5.
7. Del Rio-Chanona, E.A.; Zhang, D.; Vassiliadis, V.S. Model-based real-time optimisation of a fed-batch cyanobacterial hydrogen production process using economic model predictive control strategy. *Chem. Eng. Sci.* **2016**, *142*, 289–298. [CrossRef]
8. Cheng, Z.; Liu, X. Optimal online soft sensor for product quality monitoring in propylene polymerization process. *Neurocomputing* **2015**, *149*, 1216–1224. [CrossRef]
9. Zhong, Y.; Yang, C.; Yuchen, C.; Xuhua, S. Process real-time optimization using Clonalg algorithm. In Proceedings of the 27th Chinese Control and Decision Conference (2015 CCDC), Qingdao, China, 23–25 May 2015; pp. 743–748.
10. Moghaddam, M.; Nof, S.Y. Real-time optimization and control mechanisms for collaborative demand and capacity sharing. *Int. J. Prod. Econ.* **2016**, *171*, 495–506. [CrossRef]

11. Moksadur, R.; Avelin, A.; Kyprianidis, K.; Dahlquist, E. An Approach for Feedforward Model Predictive Control for Pulp and Paper Applications: Challenges and the Way Forward. In Proceedings of the PaperCon 2017, Minneapolis, MN, USA, 23–26 April 2017; Volume 10.
12. Optico Project. Available online: www.opticoproject.eu (accessed on 29 May 2017).
13. PROPAT-Integrated Process Control. Available online: http://pro-pat.eu/ (accessed on 29 May 2017).
14. Consens Project. Available online: http://www.consens-spire.eu/ (accessed on 29 May 2017).
15. COCOP SPIRE H2020 Project. Available online: http://www.cocop-spire.eu/ (accessed on 29 May 2017).
16. Lv, Z.; Song, H.; Basanta-Val, P.; Steed, A.; Jo, M. Next-Generation Big Data Analytics: State of the Art, Challenges, and Future Research Topics. *IEEE Trans. Ind. Inform.* **2017**, *13*, 1891–1899. [CrossRef]
17. Al-Fuqaha, A.; Guizani, M.; Mohammadi, M.; Aledhari, M.; Ayyash, M. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 2347–2376. [CrossRef]
18. Basanta-Val, P.; Audsley, N.C.; Wellings, A.J.; Gray, I.; Fernandez-Garcia, N. Architecting Time-Critical Big-Data Systems. *IEEE Trans. Big Data* **2016**, *2*, 310–324. [CrossRef]
19. Wang, K.; Fu, J.; Wang, K. SPARK—A Big Data Processing Platform for Machine Learning. In Proceedings of the 2016 International Conference on Industrial Informatics—Computing Technology, Intelligent Technology, Industrial Information Integration, Wuhan, China, 3–4 December 2016; pp. 48–51.
20. Norman, D.A.; Draper, S.W. *User Centered System Design: New Perspectives on Human-Computer Interaction*; L. Erlbaum Associates: Mahwah, NJ, USA, 1986; ISBN 0898597811.
21. Mirnig, A.G.; Meschtscherjakov, A.; Wurhofer, D.; Meneweger, T.; Tscheligi, M. A Formal Analysis of the ISO 9241-210 Definition of User Experience. In Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems—CHI EA '15, Seoul, Korea, 18–23 April 2015; ACM Press: New York, NY, USA, 2015; pp. 437–450.
22. Clark, P.G.; Lobsitz, R.M.; Shields, J.D. Documenting the evolution of an information system. In Proceedings of the IEEE National Aerospace and Electronics Conference, Dayton, OH, USA, 22–26 May 1989; pp. 1819–1826.
23. International Organization for Standardization; International Electrotechnical Commission; Institute of Electrical and Electronics Engineers; IEEE-SA Standards Board. *Systems and Software Engineering: Architecture Description*; ISO: Geneva, Switzerland, 2011; ISBN 9780738171425.
24. Rozanski, N.; Woods, E. *Software Systems Architecture: Working with Stakeholders Using Viewpoints and Perspectives*; Addison-Wesley: Boston, MA, USA, 2005; ISBN 0321112296.
25. Dzida, W. Developing Scenario-Based Requirements and Testing them for Minimum Quality. In Proceedings of the HCI International (the 8th International Conference on Human-Computer Interaction) on Human-Computer Interaction: Ergonomics and User Interfaces-Volume I-Volume I, Munich, Germany, 22–26 August 1999; Bullinge, H.-J., Ziegler, J., Eds.; Lawrence Erlbaum: Hillsdale, NJ, USA, 1999; pp. 1205–1208.
26. Chapman, P.; Clinton, J.; Kerber, R.; Khabaza, T.; Reinartz, T.; Shearer, C.; Wirth, R. Crisp-Dm 1.0. *CRISP-DM Consort.* **2000**, *76*. [CrossRef]
27. Shearer, C.; Watson, H.J.; Grecich, D.G.; Moss, L.; Adelman, S.; Hammer, K.; Herdlein, S. The CRISP-DM model: The New Blueprint for Data Mining. *J. Data Warehous.* **2000**, *5*, 13–22.
28. Pechter, R. What's PMML and what's new in PMML 4.0? *ACM SIGKDD Explor. Newsl.* **2009**, *11*, 19–25. [CrossRef]
29. Pivarski, J.; Bennett, C.; Grossman, R.L. Deploying Analytics with the Portable Format for Analytics (PFA). In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining—KDD '16, San Francisco, CA, USA, 13–17 August 2016; ACM Press: New York, NY, USA, 2016; pp. 579–588.
30. ISO 22400-1:2014—Automation Systems and Integration—Key Performance Indicators (KPIs) for Manufacturing Operations Management—Part 1: Overview, Concepts and Terminology. Available online: https://www.iso.org/standard/56847.html (accessed on 29 May 2017).