ZDMP Core Services and Middleware

Artem A. Nazarenko[a], Carlos Lopes[a], Jose Ferreira[a], Philip Usher[b] and João Sarraipa[a]

[a] *CTS, UNINOVA, DEE, Faculty of Sciences and Technology, Nova University of Lisbon, 2829-516, Monte Caparica, Portugal*
[b] *Information Catalyst for Enterprise (ICE), 37 Crewe Road, Haslington, Crewe, United Kingdom*

Abstract

Zero Defects Manufacturing Platform (ZDMP) is an initiative aimed at providing the necessary platform and tools to facilitate the transformation to the 4th industrial revolution. In this work we are addressing components that are being developed within the Work Package 5 (WP5). To suit ZDMP's objectives, the core functionalities are implemented as separate components and adjusted to the 4-tier architecture. In other words, every WP 5 component is aligned with corresponding layer: *Developer Tier*, *Platform Tier*, *Enterprise Tier* and *Edge Tier*. Moreover, every component belongs to one of the functional blocks that are also in the scope of the work: (i) Data Acquisition, (ii) Industrial Network Support, (iii) Data Harmonization, (iv) Orchestration, Monitoring and Alerting, (v) Distributed and Autonomous computing, and (vi) Analytics. At the end of the article we present a small scenario with some of the WP5 components executed at design- or run-time.

Keywords 1

Zero Defects Manufacturing Platform (ZDMP), Industrial IoT, Industry 4.0, Manufacturing Platform, vf-OS

1. Introduction

Importance of the transformation of the European industry in the context of the 4th Industrial Revolution or Industry 4.0 is difficult to overestimate. One of the reasons is the necessity to remain competitive in the highly turbulent environment of the present time, whereas being able to produce high quality products at a low cost and reduce the amount of defected products, scrap output, failures and downtime. In this regard, the concept of zero-defects in the quality management is one of the corner stone's during the transformation towards Industry 4.0, both in terms of the digitalisation of production processes, as well as the digitalisation of the product quality control.

The goal of the ZDMP (Zero Defects Manufacturing Platform – www.zdmp.eu) project is to develop and provide a digital ecosystem with the digital marketplace for connected smart factories, facilitating the compliance with the principles of the Industry 4.0. ZDMP will provide a set of tools for each step of production, while giving a possibility to extend the features using an applications store where end users will be able to add these applications as extensions to the platform according to their needs. In this context, ZDMP allows end-users to connect their systems (i.e. shop-floor and Enterprise Resource Planning systems) to acquire the platform`s functionality being able to control different stages of the production process considering time constraints and product quality. Users can also request new applications and software/hardware developers can use the ZDMP SDK (Software Development Kit) to build new Apps for them quickly using the projects toolkit and platform components. In this work we are addressing components that are being developed within the Work Package 5 (WP5). To suit ZDMP's objectives, the core functionalities are implemented as separate components and adjusted to the 4-tier architecture.

Proceedings of the Workshops of I-ESA 2020, 17-11-2020, Tarbes, France
EMAIL: a.nazarenko@campus.fct.unl.pt (A. Nazarenko); csl@uninova.pt (C. Lopes); japf@uninova.pt (J. Ferreira), philip.usher@informationcatalyst.com (P. Usher); jfss@uninova.pt (J. Sarraipa)

In terms of the usability, ZDMP will reuse outcomes / concepts of other research projects, such as: Cloud Collaborative Manufacturing Networks (C2NET), Cloud-based Rapid Elastic Manufacturing (CREMA), and especially Virtual Factory Operating System (vf-OS) which already includes more generic SDK/App. This will allow for better and faster development of the platform components in WP5-8. In addition, ZDMP will focus on the integration activities and develop suitable kick-start applications for the 4 domains pilots, as well as shape the holistic environment.

2. WP5 review

The overall architecture of the technical components of ZDMP is based on 4-tier model. The tiers are as follows: *Developer Tier*, *Platform Tier*, *Enterprise Tier* and *Edge Tier*. The main task of the Edge Tier is to enable the data collection from various heterogeneous sources such as sensors, actuators, devices and system components. It also allows some critical functionality to be delivered in the close proximity to the data sources, which might be useful in the case of asynchronous data exchange patterns. The *Enterprise Tier* is responsible for implementing the high-level communication among various users, as well as aspects related to running of the containers. This tier delivers the components for set up and maintaining the platform, whereas not providing any core functionality of the ZDMP. On the other hand, in the *Platform Tier* the core functionalities covering monitoring, alerting, high-level orchestration of the platform components and data processing are implemented. And finally, the *Developer Tier* targets particular users' needs and requirements. The components of this tier are provided as containers allowing building the complex applications comprised of containers to be later uploaded to the Marketplace.

This article targets exclusively the components developed within the WP5 of the ZDMP. To clarify which components are belonging to which tier the general view of the WP5 components in regard to the 4-tier model is presented in Figure 1. The general view of the mapping of all components developed within the ZDMP project to the 4-tier architecture can be found in [1].

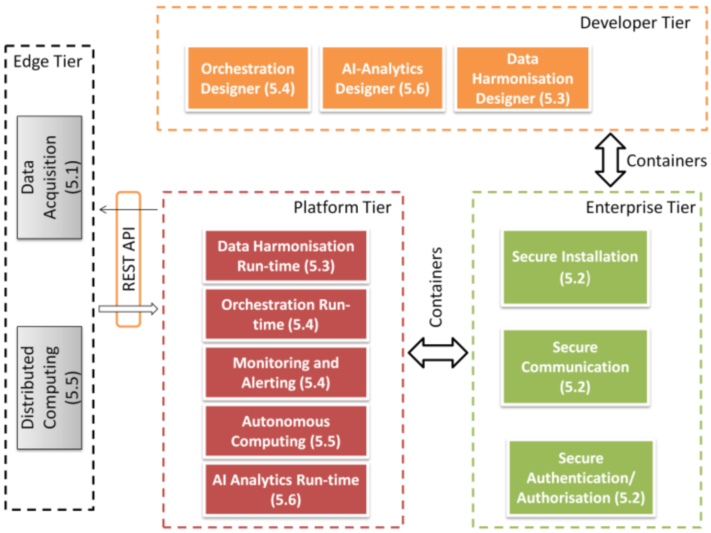

Figure 1 – High level view of the WP5 components

Important detail that needs to be mentioned is that besides the consideration of the framework through the prism of the 4-tier model, the components are also separated on run-time and design-time. Thus, in the Figure 1 the components delivered within the *Platform Tier* are implemented at run-time, whereas the components from the *Developer Tier* are created at design-time. This separation allows identifying the decision space for the entities inside the *Edge Tier* through the models established at the design-time inside the *Developer Tier*, whereas capturing the context and reflecting behavioural parameters at the run-time inside the *Platform Tier* [2].

3. WP5 review

In this chapter the core services or components of the WP5 are represented. First component is the Data Acquisition that collects data from factory's shop floor, ERP systems, and partners to assist, detect, and monitor the production process in real-time or close to real-time manner. This creates a challenge in the interoperability area where security and reliability are of main concern. Thus, ZDMP will offer the end users a set of applications to choose depending on the functionality, used communication protocol, and data source types. Data Analytics is considered as one of the key pillars in the ZDMP architecture. In this sense, the user will benefit from the reasoning applications that are offered by the platform. Additionally, and intrinsic to this feature, Big Data management will be employed to detect and/or predict any defects in the production process and parts that lead to delay or inconsistency in the delivery of the further products. The Resource Orchestration is provided to act for avoiding any defects in the production process. In addition, assessment will be provided by the platform on product quality assurance in case there is misalignment with the product quality standards. Further, the orchestration takes the human into account, where collaboration is needed in manual production systems.

The WP5 is divided into six components that are described in the next sub-chapters.

3.1. Data Acquisition and IIoT

Data Acquisition and IIoT component delivers the main input to the platform. It provides an interface with a number of external sources to gather data. The external sources can be represented as industrial automation devices (PLCs, smart sensors, RFID readers, etc), ERP systems, SCADA/MES systems/data, and existing databases. The component offers a set of interfaces to bridge the Platform Tier with heterogeneous sources, whereas providing the abstraction level to hide the communication complexity from the other components. This component has the following main elements:

- **Data Source Adapters** are intended to perform the read/write operations and manage configuration, status and command data of the end-devices. Each Adapter bridges the end-devices with the Data Source Manger that contains the relevant configuration data. Moreover, this component contributes to interoperability by enabling conversion from the device-specific protocols to the formats utilized within ZDMP. Both synchronous and asynchronous data transmission patterns are considered.
- **Data Source Manager** is responsible for managing of end-device configuration and metadata, and gathering the logs.
- **Configuration UI**, the main goal of this component is to allow the user to access and change the configuration data of the end-devices.
- **Data Source Registry:** is the repository for the configuration data.
- **Variety of APIs:** the component provides a set of APIs including Metadata API, Logging API, Security API, Data Source Query API and Data Source Gateway API. The last one provides on the spot an access to configuration and end-device data for the rest of the platform, whereas directly supplying the Harmonisation module with necessary data.

This component utilizes some deliverables based on previous progress Cumulocity IoT[2] and the vf-OS IO toolkit[3]. This will allow the connection via various protocols (eg OPC-UA[4], OMA Lightweight Machine to Machine[5], IPSO Smart Objects[6]).

3.2. Robust Industrial Network Support

[2] https://www.softwareag.cloud/site/product/cumulocity-iot.html#/
[3] https://opensourceprojects.eu/p/vfos/iotoolkit/wiki/IO%20Toolkit/
[4] https://opcfoundation.org/about/opc-technologies/opc-ua/
[5] https://www.omaspecworks.org/what-is-oma-specworks/iot/lightweight-m2m-lwm2m/
[6] https://www.omaspecworks.org/develop-with-oma-specworks/ipso-smart-objects/

This component focuses on security and privacy aspects of ZDMP. The risks mitigation is assured through a variety of measures [3], such as minimization of attack surface by isolation and zoning of communications, implementation of robust encryption, enforcement of access control policies, authentication of entities and introduction of security monitoring entities. In other words, the particular component is intended to cover relevant issues around security and privacy domain. The core elements of this component are: Secure Installation, Secure Communication and Secure Authentication/Authorization [4].

Secure Installation facilitates the secure installation of zApps from the marketplace to the platform at the run-time. This component also manages trust and legitimacy of any installed zApp and creates users, roles and policies required by the zApp.

Secure Communication provides a public key infrastructure for other ZDMP components. This element enables secure communication across the platform. Moreover, it includes certification and registration authorities providing the core functionality of the sub-component – issuing, revoking certificates and matching identification with certificates.

Secure Authentication/Authorization is responsible for authorization and authentication within the platform. This sub-component checks the accordance of the requests to the specific access policies and in the case of the positive assessment issues the access token. It also makes use of the best practices, while implementing the OpenID[7] and the OAuth2.0[8] protocols to protect resources from unauthorized access.

These three security sub-components are linked closely with the key functionality of the platform, namely with the Service and Message Bus, Portal, Application Runtime and Marketplace. This protects from some critical vulnerabilities and allows security to be applied to the whole platform.

3.3. Data Harmonization and Interoperability

The Data Harmonisation and Interoperability component is based upon and is further development of the Semantic Management Component [5]. Semantic Management Component is composed of two modules the Harmonisation and the Storage. Harmonisation module is responsible for communication with the Storage component, data cleaning and cross-format adjusting of the input files and detection of the data patterns in the row data. The Storage module allows storing the ontologies and integrating the newly imported ontologies to already existing Knowledge Base. The task of harmonization and interoperability provision is split into the run- and design-time blocks:

The **Data Harmonization Designer,** data transformation is one of the tasks of this component, adjusting the data formats to the needs of requesting entity. Another task is the semantic homogenization supporting the data mapping through ZDMP Knowledge Base. Further task of the component implies the mapping of the users' or applications' data and metadata to the platform reference model. Moreover, the component provides the necessary mechanisms to the user to create Manufacturing Maps containing the rules for transforming the specific types of syntax.

The **Data Harmonization Run-time** assures the proper deployment, and encapsulation of the Data Harmonization Designer. Afterwards, the functionality of the Data Harmonization Designer is available as a service at the Marketplace. The procedure follows the ETL workflow, whereas extracting, transforming and loading the data. This is based, in particular, on the Talend Open Studio for Data Integration.

3.4. Orchestration, Monitoring, and Alerting

The component that is responsible for the orchestration is divided into two parts, namely the one which is executed at the run-time – Orchestration Run-time, and the other at the design time – Orchestration Designer. The approach is similar to the one used in vf-OS project for harmonization component [6], where data mapping is accomplished at the design-time and data transformation at the run-time.

[7] https://openid.net/
[8] https://oauth.net/2/

The **Orchestration Designer** assists processes management using the business process model notation (BPMN[9]). The BPMN models allow visual representation of a sequence of events for the business modelling. Another important task of this sub-component is to map different file formats. In other words, if the file format is not suitable to the consuming component's acceptable format, it is changed to fit it. Furthermore, the sub-component contributes to the improvement of suggestion quality of the user's requests, whereas effective mappings are added to the knowledge base. It also supports the ontology storage, keeping the relevant knowledge in the graph form [5], which is useful for analysis of relations among the concepts.

The **Orchestration Run-time** interprets and implements process flows delivered from the Orchestration Designer. To do this, it connects to the required services and executes them according to the logic described in the BPMN diagram. For this purpose the sub-component possesses the BPMN parser and BPMN map API.

The **Monitoring and Alerting** This component is used to collect the data from the end-devices, which later are represented in the graphical form in a web-based user interface. It might be also used for historic data and logs collection. The second function is to generate alerts, based on data received and inform all involved counter-parts. The alert can be sent in various ways, such as, email, HTTP endpoints or SMS.

3.5. Distributed and Autonomous Computing

Distributed Computing provides mechanisms for applying the distributed computing technologies, such as: fog, edge and mist environments. One of the key issues being in focus of this component is the ability to efficiently divide the task onto smaller ones and distribution among available nodes for further execution. This includes many aspects, such as managing the distribution, mapping the locations of the computational nodes, load balancing and API to communicate with other components.

The **Autonomous Computing** component covers the issues related to the processes automatisation, as well as communication automatisation among other components. The key elements of it are:

- User Interface for setting the policies – on this stage the rules to activate the triggers are set. Crucial is to specify "how each user may change the policy" [7] in order to avoid and resolve conflicts, and predict the consequences of applying the certain policy.
- Triggers for autonomous processing – following the reactive paradigm, the triggers are activated depending on the data obtained.

3.6. AI and Analytics

The **AI-Analytics Designer** component enables creation of machine learning models from the historical data, which later are run by the AI-Analytics Run-time [4]. The main goal is to detect and predict defects in the production process. For this purpose supervised as well as unsupervised learning algorithms are applied.

The **AI-Analytics Run-time** is used for deploying the models on the platform and run them in synchronous or asynchronous manner. Visualisation is also in the scope of this component as a user interface with various dashboards. It detects the anomalous behaviours through analysis of measurements and identification of deviations from the learning models delivered from AI-Analytics Designer.

4. Possible Application Example

The components belonging to the WP5 are presented with accordance to 4-tier model and in terms of functionality are divided into those that executed at the design- and the run-time. The basis for

[9] http://www.bpmn.org/

some of the components is being developed within some passed projects. A good example is the *Data Harmonisation Designer*, which was designed and developed within the vf-OS project [5]. However, the basis will be enriched with more services to be consumed by the other components, such as *AI-Analytics* (Fig. 2). In Fig. 2 the small scenario combining the *AI-Designer*, *Semantic Management Component* which is the vf-OS analogue of *Harmonisation Designer* and *Data Acquisition Component* is presented. All presented components are separated according to the execution at the run- or the design-time. Data Acquisition component is responsible for the data flows management from devices to other components and in particular to the *Harmonisation Designer*. The user that in the same time can have the designer role can change the configuration through appropriate interface provided by *Acquisition* component. The *AI-Designer* offers to the *Harmonisation Designer* a set of algorithms for data enrichment or feature extraction, which can be accessed on demand. And finally the *Data Harmonisation Designer* is represented in detail, including harmonisation part itself and the ontology storage. Among the core deliverables of it are:

- Supplying the user with helpful suggestions when performing mappings between distinct concept knowledge bases;
- Storage of ontologies with support of various data formats;
- Transformation of generic requests into semantic queries;
- Interface to manage the access to component enabling the reception of different file formats and thus contributing to interoperability;
- Import and integration of third-parties ontologies.

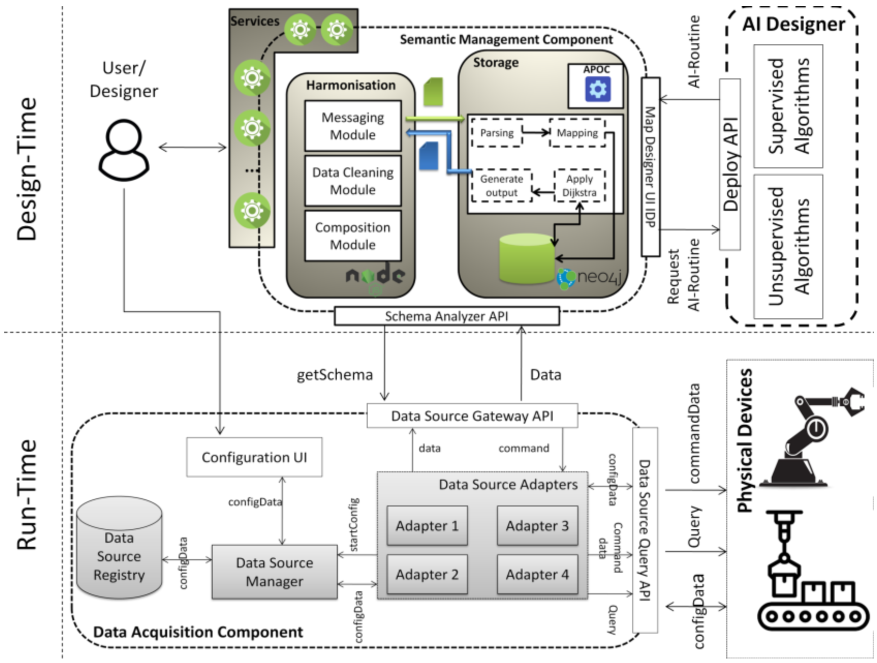

Figure 2 – Small scenario involving some WP5 components

5. Conclusion

The current research paper aims at giving the overview of the WP5 components being developed within the framework of ZDMP project. This work package for Core Services and Middleware is represented through key functional blocks: (i) Data Acquisition, (ii) Industrial Network Support, (iii) Data Harmonization, (iv) Orchestration, Monitoring and Alerting, (v) Distributed and Autonomous computing, and (vi) AI Analytics. The ZDMP platform is based on 4-Tier model and thus all the components are represented in regard to it. Moreover, the functionality of most of the components is divided onto two parts, the one which is executed at the design-time and the other being executed at the run-time. The components within the Platform tier belong to the run-time, whereas components within the Developer tier to the design-tier category. This allows considering separately the design of the models and the execution of the models developed during the design phase. Important to mention, that components are designed in the modular way in order to ensure the flexibility to add or remove the needed functionality. Further work includes the finalisation and adjustment of the components being developed within the WP5 with the other components of ZDM platform followed by the application in the real-case testing scenario.

6. Acknowledgement

The research leading to these results received funding from the European Union H2020 Program under grant agreement No. 825631 “Zero Defect Manufacturing Platform (ZDMP)”.

We thank Perez Salvador (IKERLAN, Spain), Mircea Vasile (SIVECO, Romania), Laura Caroline Ribeiro de Melo (Ascora GMBH, Germany) and Christian Melchiorre (Softeco Sismat S.r.L, Italy) for the valuable assistance in designing of the framework’s architecture.

7. References

[1] F. Fraile, R. Sanchis, R. Poler, A. Ortiz, Reference Models for Digital Manufacturing Platforms. Appl. Sci., 9, 4433, 2019. https://doi.org/10.3390/app9204433

[2] A. Steck and C. Schlegel, Managing Execution Variants in Task Coordination by Exploiting Design-Time Models at Run-Time. IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, USA, 25-30 September 2011, p. 2064-2069. https://doi.org/10.1109/IROS.2011.6094732

[3] T. Dellas, WP4: Technical Challenge: Requirements, Specifications, and Standardisation. D4.4a: Functional Specification – Vs: 1.0.3., 2020. URL: https://c53c19bc-6460-4dea-a74f-97b851e7af75.filesusr.com/ugd/9ebc7c_2e7bb64dbb30473ab42ba6ff55b9efd6.pdf

[4] P. Usher and J. Tryand, WP4: Technical Challenge: Requirements, Specifications, and Standardisation. D4.3a: Global Architecture Specification – Vs: 1.0.0., 2019. URL: https://c53c19bc-6460-4dea-a74f-97b851e7af75.filesusr.com/ugd/851c99_0205c297a4b547bf8665daa36cf70984.pdf

[5] A.A. Nazarenko, J. Sarraipa, L.M. Camarinha-Matos, O. Garcia and R. Jardim-Goncalves, Semantic Data Management for a Virtual Factory Collaborative Environment, Applied Sciences, 9, 22, (4936), 2019. https://doi.org/10.3390/app9224936

[6] A.A. Nazarenko, J. Giao, J. Sarraipa, O.J. Saiz, O. Garcia Perales and R. Jardim-Gonçalves, (2018). Data Management Component for Virtual Factories Systems. In Enterprise Interoperability: Smart Services and Business Impact of Enterprise Interoperability; Zelm, M., Jaekel, F.-W., Doumeingts, G., Wollschlaeger, M., Eds.; ISTE Ltd.: London, UK; pp. 99–106, 2018. https://doi.org/10.1002/9781119564034.ch12

[7] P. Gupta, S. D. Stoller and Z. Xu, "Abductive Analysis of Administrative Policies in Rule-Based Access Control," in IEEE Transactions on Dependable and Secure Computing, vol. 11, no. 5, pp. 412-424, 2014. https://doi.org/10.1109/TDSC.2013.42