2018 Sixth International Conference on Enterprise Systems

Specification of a Software Architecture for an Industry 4.0 Environment

Evangelia Kavakli*†, Jorge Buenabad-Chávez*, Vasilios Tountopoulos‡, Pericles Loucopoulos*, Rizos Sakellariou*
*School of Computer Science, The University of Manchester, Manchester, UK
†Department of Cultural Technology and Communication, University of the Aegean, Mitilini, Greece
‡Athens Technology Center S.A., Chalandri, Greece
{evangelia.kavakli-2,jorge,pericles.loucopoulos,rizos}@manchester.ac.uk,v.tountopoulos@atc.gr

***Abstract*—Data-driven decision making is at the core of Industry 4.0. This paper describes the specification of a conceptual architecture of a smart system for supporting decision making in the context of disruptive events in manufacturing operations. Following a viewpoint-oriented approach, the proposed architecture identifies the functional components that facilitate decision making and establishes the interfaces between them, demonstrates the information flow within the manufacturing ecosystem for vertical / horizontal integration and establishes the mapping of the functional components to different software containers, execution environments and physical devices.**

***Index Terms*—Smart manufacturing, software architecture, decision making, Industry 4.0.**

I. Introduction

The term Industry 4.0 has recently been adopted to refer to the technologies that aim to deliver the next significant transformational change in the organization and management of industrial processes [1]–[3]. Building upon a range of technologies, such as Cyber-Physical Systems (CPS) [4], Internet of Things (IoT) [5], cloud computing [6], and big data analytics [7], the aim is to take full advantage of the abundance of data in modern manufacturing enterprises to extract information that helps make ‘smart’ decisions leading to the so-called *smart manufacturing* [8], [9].

At the core of smart manufacturing is the integration of data from different sources and different software technologies. As a matter of fact, the deployment of any data-driven decision making system into the manufacturing domain involves the integration of computational elements with sensors, devices, equipment, and systems found within manufacturing facilities along with legacy enterprise information systems and advanced software such as schedulers or optimizers. It is evident that this multitude of different and heterogeneous resources and data creates a *system-of-systems integration challenge* [10].

A key element in dealing with this challenge is architectural design [11]. In fact, the architecture is a key facet of the design of any software-intensive complex system. It includes its fundamental organization embodied in its constituents, their relationships to each other (constituent) and to the environment, and the principles guiding the system’s design and evolution [12].

This paper reports the specification of a software system architecture to support decision making for managing disruptive events in smart manufacturing. The architecture builds upon widely accepted system architecture practices as well as emerging trends reported in Reference Architectures for smart manufacturing. Following established specification methodologies, the proposed system architecture consolidates and generalizes the requirements of two concrete business cases in the automotive and the white goods sectors and has been developed in the context of the EU-funded H2020 project DISRUPT[1]. The aim of DISRUPT is to facilitate the transition into Industry 4.0 smart factories through a data-driven, flexible system that supports decision making and enactment of decisions when events that disrupt enterprise operations, such as delays in the supply chain or failures in a production line, may occur [13]–[15].

The paper is structured as follows. Section II, provides an overview of related work. The detailed description of the proposed architecture is provided in Section III and its application on supporting smart decision making in Section IV. Finally, Section V concludes and provides pointers to future developments.

II. Background and Related Work

Smart manufacturing systems are networked information systems that are tightly coupled with the physical processes. The development of such systems requires the collaboration of different engineering disciplines in addition to software engineering, such as mechanical engineering, electrical engineering, etc. using diverse representation schemes, having diverse domain knowledge and different development strategies.

To deal with the complexity of system development in a setting with many actors, system architecture practice embraces the concept of architectural *viewpoints*, whereby the architectural description is partitioned into a number of separate (though complementary) representations, each reflecting specific concerns held by one or more of its stakeholders, using certain conventions such as notations, modelling methods, analysis techniques etc. In fact, several architecture frameworks have been proposed, which are essentially viewpoint

[1]http://www.disrupt-project.eu/

classification schemes. The classification by Kruchten [16], proposed in 1995, is probably the best known classification and is still widely used. The viewpoint-oriented paradigm has been adopted by ISO/IEC 42010:2007, later revised by the ISO/IEC 42010:2011 Standard "Systems and software engineering–architecture description" [17].

The concept of viewpoints is widely used across the enterprise systems community and appears in several enterprise modelling frameworks including the Enterprise Knowledge Development Framework [18], the Multi-Perspective Enterprise Modeling (MEMO) [19] and the Open Group TOGAF architectural framework [20], among others.

Similarly, in the context of manufacturing enterprise systems, the Architecture Integrated Information Systems (ARIS) [21], the Computer Integrated Manufacturing Open System Architecture (CIMOSA) [22] as well as the more generic Generalised Enterprise Reference Architecture and Methodology (GERAM) [23] also consider the development of manufacturing information systems from a number of views (organizational, functional, informational), aiming to facilitate the alignment between a company's strategy (the business domain) and its supporting IT systems as well as between IT applications across the enterprise (system integration).

The design of intelligent manufacturing systems has also been considered in the context of cyber-physical production systems engineering [24]. Research in this area has focused on Architecture Description Languages (ADLs) for supporting model-based design of CPSs, ranging from semi-formal descriptions e.g. SysML [25] to formal descriptions e.g. SosADL based on a p-Calculus [10]. Their main focus is in the provision of a common language for describing how various parts of CPS are constructed and how they operate. Such ADLs can be used to represent architectural specifications, however unlike the proposed approach they do not define any specific viewpoints to accommodate the requirements of relevant stakeholders.

The advancement of the Internet of Things and its application to the manufacturing industry has brought forward new architectural models, including the Industrial Internet Reference Architecture (IIRA) [26], the Draft ISO Internet of Things Reference Architecture (IoT RA) [27] and the Reference architecture model Industrie 4.0 (RAMI 4.0) [28]. A range of efforts to develop different software architectures in specific contexts in relation to Industry 4.0 is reported in [29]–[32].

The characteristic of these reference architectures is that they are expanded in scope to consider the convergence of IT systems with the operational (physical) domain. In particular, they consider manufacturing systems as cyber-physical systems comprising interacting physical and digital components. To this end, they involve viewpoints and concepts related to business, applications and IT infrastructure, but also to the physical environment.

In addition, a common approach used to address the increasing complexity of smart manufacturing systems is the separation of concerns through modularization and decomposition of the system into interdependent functional components, with distinct capabilities that can be realized by one or more implementations of actual system components. To address composition and interoperability purposes the component description should also include the definition of the interfaces and services made available by each component.

Following the above design principles the proposed architecture consists of loosely coupled components which closely match the IIRA functional domains, specified using a fixed set of viewpoints as described in Section III-B.

III. DISRUPT Software Architecture

A. DISRUPT requirements

The architecture vision of DISRUPT is guided by the desire to support knowledge-driven decision making in manufacturing enterprises through the efficient identification and handling of events in the manufacturing ecosystem that could disrupt enterprise operations, as, for example, delays in the supply chain (horizontal integration) or failures in a production line (vertical integration).

As such, DISRUPT aims to address the following functional requirements:

- To collect and aggregate multi-source, multi-scale and multi-variant data, which might be stored into existing Enterprise Information Systems or captured on the deployed IoT infrastructure.
- To detect actual disruptive events or identify trends that may disrupt manufacturing and supply chain operations based on the analysis of collected data.
- To establish situation awareness of concerned stakeholders (e.g. shop-floor managers, logistics managers, etc.), estimating the impact of an event on the performance of the manufacturing operations at different time scales.
- To manage the decision making process to address the observed or expected disruptions, through the exploitation of existing knowledge for generating alternative courses of action, as well as the simulation alternatives, optimising them to meet specific objectives.
- To actualise informed decisions, by notifying the selected decisions to the relevant stakeholders, as well as by enforcing the automatic actualization of decisions towards low level physical components.

In addition, the following non-functional concerns are considered fundamental to the architecture of the system:

- The system should support, in close to real-time, decision making and enactment of made decisions under events that disrupt its operations.
- The system will be added onto an existing factory ecosystem.
- The system will be based on a distributed architecture, consisting of a loose coupling of (potentially existing) functional components, to enable flexibility in the system components and in the functionality, thus reducing the risk for a vendor lock-in.

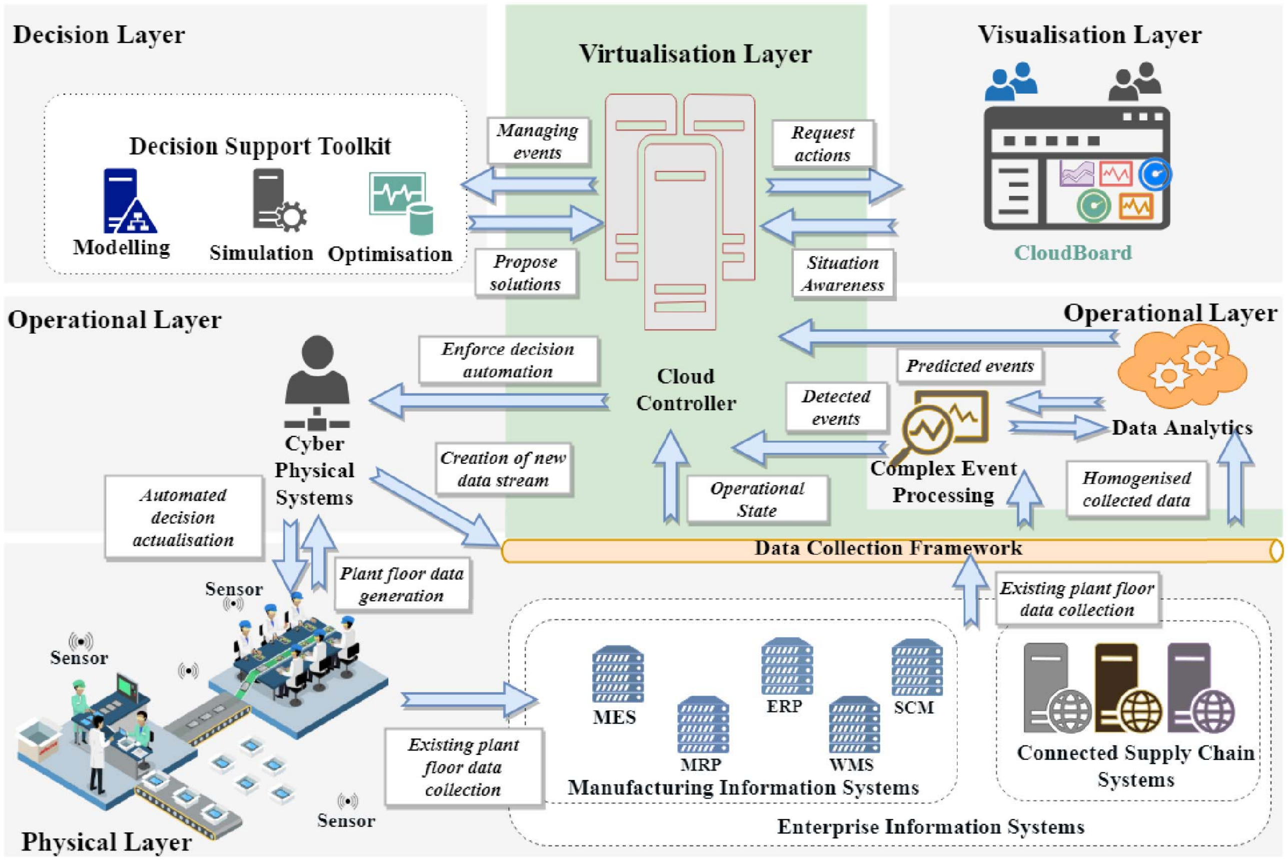

Fig. 1. DISRUPT system components

Figure 1 provides an overview of the DISRUPT system components and how these interact in order to provide the required functionality. These components are organised into five layers each having a specific responsibility, examined in more detail in the following paragraphs.

In particular, on the *Physical Layer*, the current production layout of the manufacturing organisation exists, which consists of factory assets that produce data. These assets can be machines, lines, sensors, etc., which feed the existing Manufacturing Information Systems (MIS) with data regarding the execution of a production planning, the use of materials, the production of (semi-)finished products and any other operation that runs in a factory setting (depending on the factory under consideration). The Physical Layer also comprises the infrastructure to generate sensor data, to be exploited by the **Cyber Physical Systems (CPS)** component residing in the operational layer.

On the *Virtualisation Layer*, the **Data Collection Framework (DCF)** component is the DISRUPT interface to the Physical Layer. As such it aggregates the multi-source, multi-scale and multi-variant data from the Enterprise Information Systems residing in the Physical Layer, as well as the CPS and prepares it for processing and analysis, by transforming them into an internal to DISRUPT homogeneous data structure. This layer, also includes the **Cloud Controller**, which is the heart of the DISRUPT platform. It interfaces between DISRUPT components and manages communication between them.

On the *Operational Layer*, the homogenised collected data is analysed by the **Complex Event Processing (CEP)** component to detect disruptive events that should be communicated to the Cloud Controller. In addition, the **Data Analytics** component exploits the homogenised collected data to identify trends or patterns that could offer insights on events that might occur in the future, based on the current operational status of the factory (and the implementation of a selected production plan) and the involved assets. Eventually, the events produced by the Data Analytics component could be exploited to configure CEP for monitoring specific rules in the future. On the other hand, the CPS interacts with the shop floor and produces real time data which is collected through the DCF and fed into CEP, to detect disruptive events and communicate these events to the Cloud Controller. Also, the Cloud Controller can send CPS an order for automated actions to be performed at runtime, which in turn the CPS propagates to the assets of the Physical Layer.

On the *Decision Layer* the **Decision Support Toolkit (DST)**, consisting of three internal components, namely **Modelling**, **Simulation** and **Optimisation**, enables the modelling of the manufacturing knowledge, as per the organisational environment, manufacturing and decision processes, upon which the handling of disruptions is based. Furthermore, it is responsible for analysing the impact of disruptions due to the existence of one or more events and proposing, upon request (by the end-user or generated by the Cloud Controller automatically), a set of solutions to address these disruptions. Through the Cloud Controller these solutions are communicated to the entities (human actors or automated agents), responsible for taking the decision.

Finally, on the *Visualisation Layer* the **CloudBoard** component is the interface between the DISRUPT platform and its end users. In particular, it is responsible for informing the end users about what is actually happening in the various manufacturing operations (situation awareness). Through this component, end-users are able to monitor the different variables affecting the manufacturing processes, invoke the DST and enact decisions on the appropriate solution(s) to handle disruptions (request action). Although the candidate solutions to address these disruptions are produced in the Decision Layer, their actualisation may be attributed to the Visualisation Layer (to notify the involved end-users) and be shaped through the CPS in the Operational Layer.

B. *DISRUPT Architecture Viewpoints*

The proposed architecture specification focuses on the identification of the system functional components, the interaction between them in order to achieve the intended system behaviour, and their deployment into the operational environment of an existing manufacturing organization. The DISRUPT architecture viewpoints have been defined by analysing the specific requirements of the two industrial partners (described in Section III-A), identifying the relevant stakeholders of the DISRUPT system and determining the proper framing of concerns. In particular, three architectural viewpoints have been considered: *logical*, *informational* and *physical* viewpoints [14].

The logical viewpoint describes the high level structure of the architecture in terms of the functional components that collaboratively deliver the required functionality. The emphasis is on the relationships between the components realized through external interfaces and not the internal structure of components. The concerned stakeholders are the system users (domain experts, operational managers and decision makers), the system administrators, as well as the technology providers (including the suppliers, the developers, the integrators, the testers and the maintainers of the system). Suitable model kinds for this viewpoint include UML class diagrams and UML component diagrams.

The informational viewpoint focuses on the dynamic behavior of the system, in terms of the runtime information flow between system components. Relevant stakeholders are mainly the system users and technology providers. UML informational flow diagrams can be used to model circulation of information at a high level of abstraction, whilst UML sequence diagrams to describe the exchange of information between components at a certain level of detail.

Finally, the physical viewpoint details the allocation of the system components to different software containers, execution environments and physical devices. Typical concerns framed by this viewpoint include the types of hardware required, network requirements, and physical constraints. Concerned stakeholders are the system administrators, as well as technology providers. Appropriate model kinds for this viewpoint are UML deployment diagrams.

IV. The DISRUPT Architecture for Managing Decision Making

This section provides a detailed description of the DISRUPT architecture through the viewpoints specified in Section III.

A. *Logical architecture*

Figure 2 shows the DISRUPT logical architecture in terms of the logical structure of DISRUPT components and their interconnections. For each component, this structure presents the main functional blocks as UML classes, including the involved DISRUPT data descriptions and the associated methods, which exploit this data to produce the expected outcome. The latter can be made available to other components through interfaces. These interfaces are also described as UML class diagrams in Figure 2, labeled 'interfaces', in which the main interface methods are provided.

The following list provides an overview of the main DISRUPT components and their interfaces:

- The Cloud Controller holds a central position. This component implements four functional blocks that implement the main responsibilities of the Cloud Controller in the DISRUPT methodology. Apart from the internal functions, the Cloud Controller exposes four interface classes that enable the communication of this component with the CloudBoard (UserData and INotification Interfaces), the components in the Decision Support Toolkit (IDSSConfig Interface) and the Cyber-Physical System component (IActionHandler Interface).
- The Data Collection Framework is a platform infrastructure component that distinguishes between two main functional sub-components. The first one relates to the Data Collector class that is a Message Bus implementation, exposing the interfaces for publishing streaming data (IPublish Interface) and subscribing for respective information (ISubscribe Interface). The second one relates to the batch data acquisition process and exposes an interface for connecting with the existing information systems (IGetData Interface).
- Relevant to event analysis, DISRUPT offers two different components, one relating to batch analytics (the Analytics component) and another one that relates to streaming analytics (the Complex Event Processing component). The Analytics component provides two UML classes that implement the functionalities for the batch analytics process to train data, based on models, and the prediction

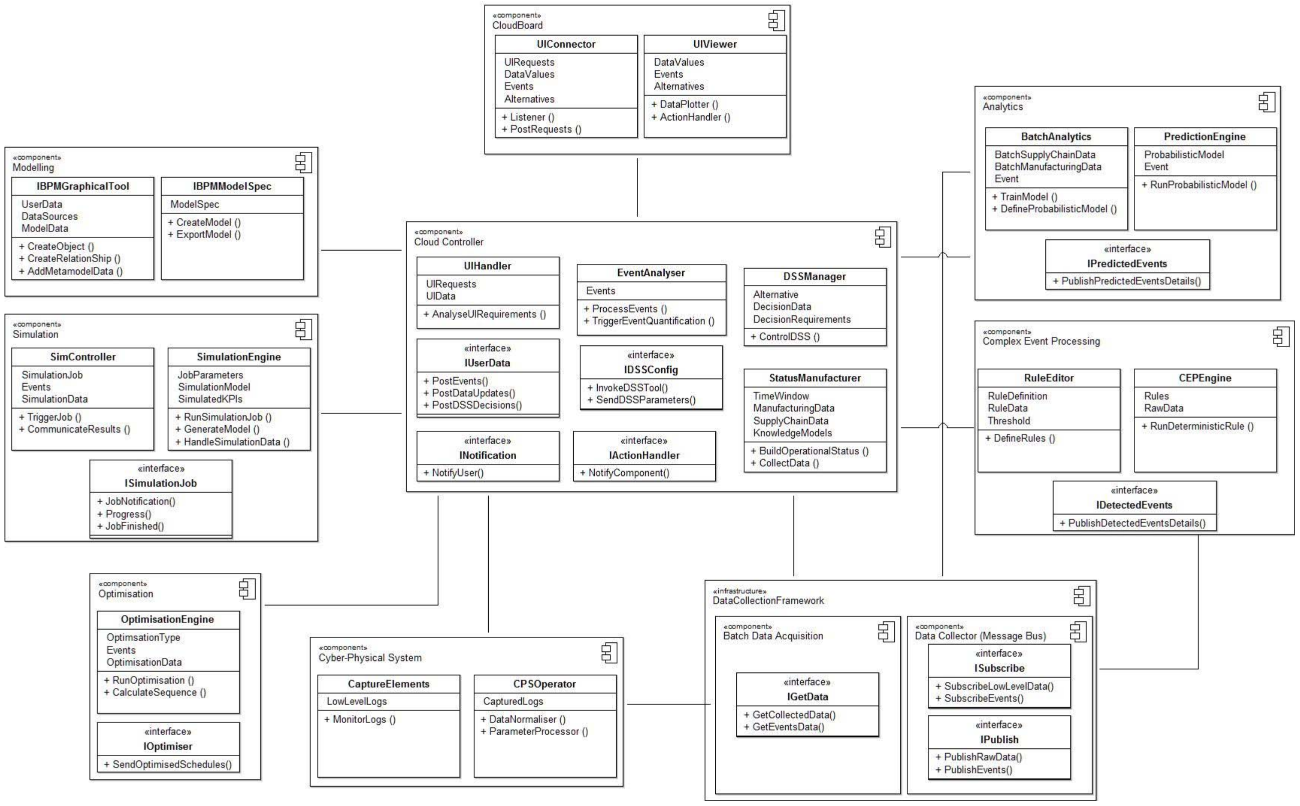

Fig. 2. The DISRUPT logical architecture.

engine, which runs the predictive analytics models for event prediction. This component implements an interface for publishing the details of the predicted events mainly to the Cloud Controller (IPredictedEvents Interface). On the other hand, Complex Event Processing defines an internal structural behaviour that detects events, based on deterministic rules and exposes an interface for publishing the details of the actual events mainly to the Cloud Controller (IDetectedEvents Interface).

- The Decision Support Toolkit component (consisting of Modelling, Simulation and Optimisation) uses the Cloud Controller to orchestrate the decision making process. The Decision Support Toolkit subcomponents implement their internal logical structure, while they expose separate interfaces to communicate their results to the Cloud Controller. As such, the Simulation subcomponent exposes an interface to control and manage the simulation jobs (ISimulationJob Interface) and the Optimisation subcomponent implements the interface to run optimizations on the production planning and scheduling (IOptimiser Interface).

B. *Informational architecture*

Moving to the details of components' interactions, the DISRUPT informational architecture depicted in Figure 3 demonstrates how data travels within the DISRUPT environment, starting with the collection of manufacturing and supply chain data from existing Enterprise Information Systems and IoT devices. In that respect, it describes the relationship of each component with the DISRUPT datasets.

In more detail, shown at the top right part of Figure 3, the Data Collection Framework and Cyber-Physical System components retrieve disperse data from the various enterprise operations and process them to generate alerts coming from detected events (through the Complex Event Processing component) or trends predicted from data dynamics (in the Analytics component). Both types of events and the related multiscale and multisource datasets empower human agents in understanding what is happening in the manufacturing ecosystem, and realise the impact of the disruptive events on the enterprise operational status and the defined business goals and Key Performance Indicators (using the CloudBoard). Furthemore, through modern simulation and optimization techniques, DISRUPT supports recommendations (alternative plans) for handling potential disruptions and enhancing decision making through informed choices, which can be used by the Cloud Controller to support the enforcement of actions both in a manual (through the CloudBoard) or automated manner (through the Cyber-Physical System component) across the manufacturing ecosystem. Finally, shown at the top left corner of Figure 3, the DISRUPT Modelling component, enables the production of models, representing the knowledge of the manufacturing

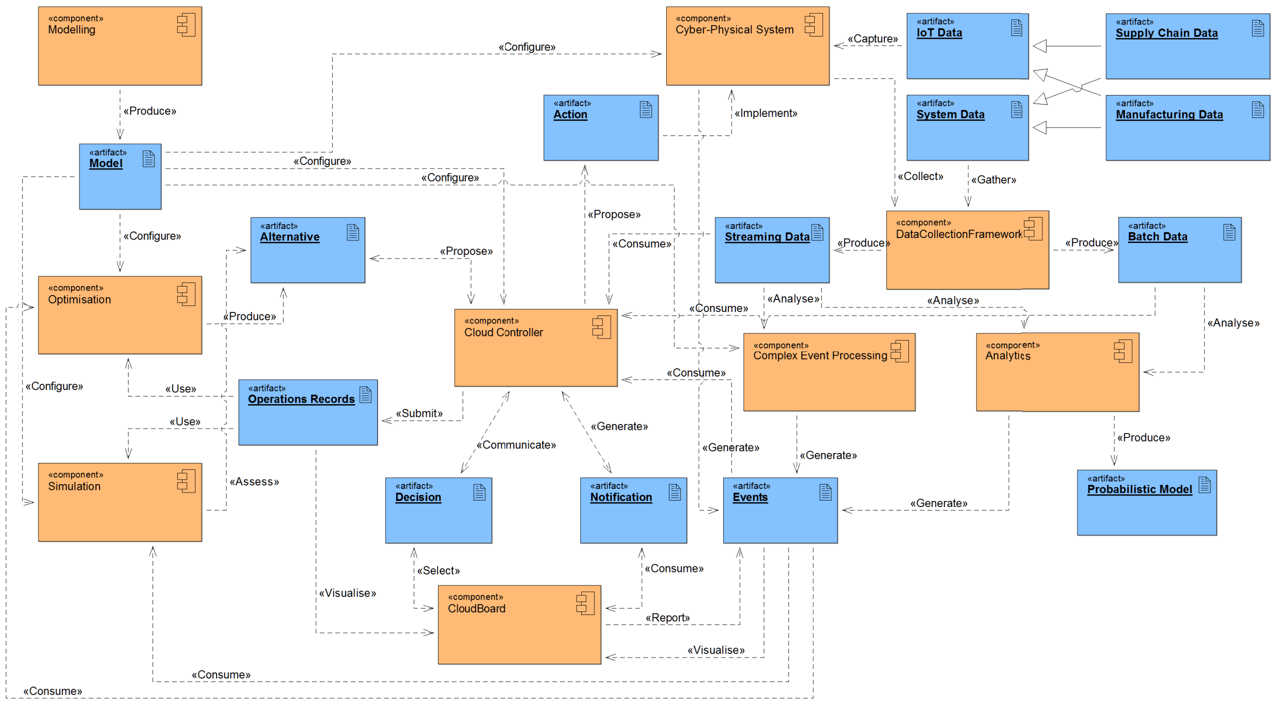

Fig. 3. The DISRUPT overall informational architecture.

production and supply chain processes across different layers. These models can configure the Cloud Controller at runtime and be used in the Simulation and Optimisation components to govern their operations, according to this knowledge.

In addition, sequence diagrams are used to further detail the flow of information and the interaction between the DISRUPT components in the context of the functions described in Section III-B. For example, Figure 4 summarises the components' interactions that refer to the management of decision making.

The DISRUPT decision making process aims to assist the stakeholders in making informed decisions on how to tackle disruptive actual and/or predicted events in their manufacturing and supply chain operations. As shown in Figure 4, the CloudBoard allows the involved stakeholders to request for support with disruptive events. This request is attributed to the Cloud Controller, which, in turn, decides which Decision Support component will be activated to examine the alternative solutions. The decision making process includes a series of interactions between the Cloud Controller and the Decision Support components (especially Simulation and Optimisation) on a case by case basis.

In Figure 4, the alternatives may be instructed directly from the knowledge model to the Cloud Controller or the latter may ask the Optimisation component to produce a new list. In both cases, the Cloud Controller is responsible for invoking different simulation jobs in the Simulation component to evaluate whether an alternative can address the impact raised by the events. The goal is that, eventually, the Cloud Controller receives (and later forwards) the necessary information that would empower the DISRUPT end-users in making an informed decision about handling the potential disruptions in the respective operations. Once the list of alternative actions is available, along with the estimation of their impact on the current production plan, this list is sent to the Cloud Controller and is presented to the stakeholders through the CloudBoard.

Similarly, the sequence diagram of Figure 5 details the flow of information and the interaction between the DISRUPT components with respect to the actualisation of the decisions. In particular, once a user decides which alternative to implement, the CloudBoard communicates it to the Cloud Controller. The latter must reason between: (a) notification decisions, in which the Cloud Controller is responsible for sending the decision selected on the CloudBoard to another stakeholder; and, (b) the potential enforcement of automatic actualization, in which the decisions are applied by CPS at runtime to the assets of the Physical Layer.

C. Physical architecture

The physical view details the allocation of the DISRUPT components to different software containers, execution environments and physical devices. As shown in Figure 6, the DISRUPT platform resides on a server-side environment, which is distributed and hosts the processing and managing components for making decisions, as separate containers. To this end, we identify containers for: (i) the data collection and data acquisition processes, which are implemented through

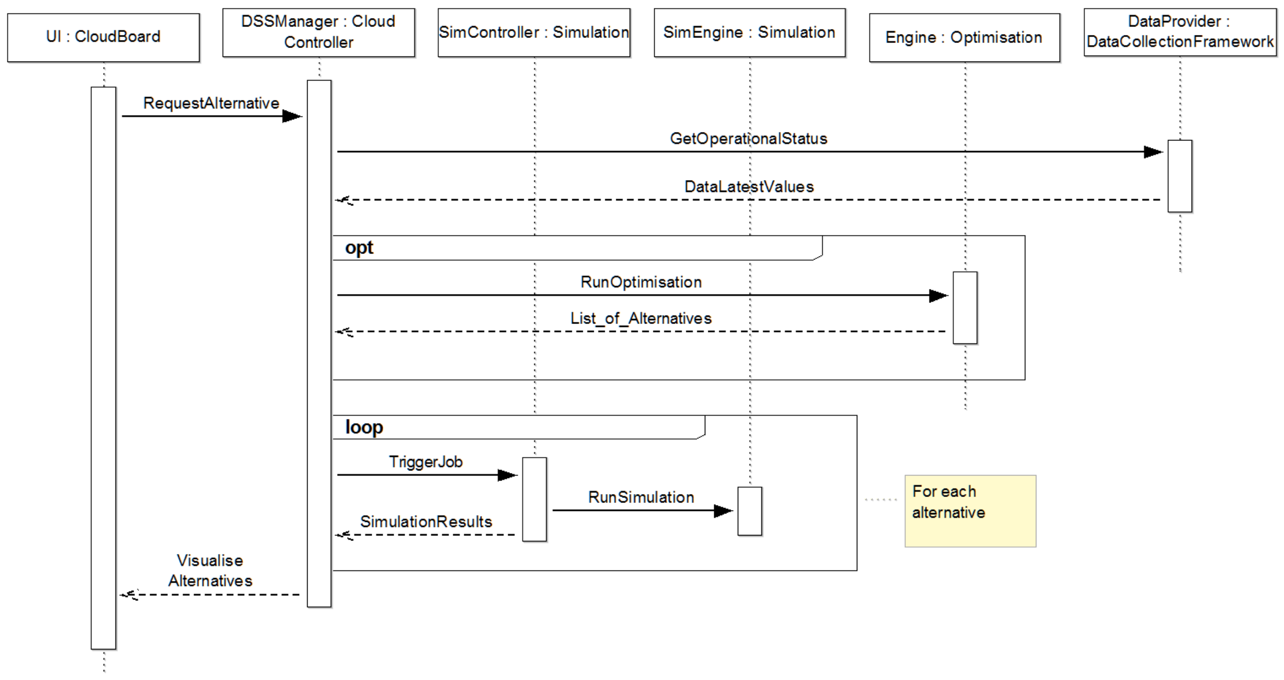

Fig. 4. Information flows and component interaction for managing decision making.

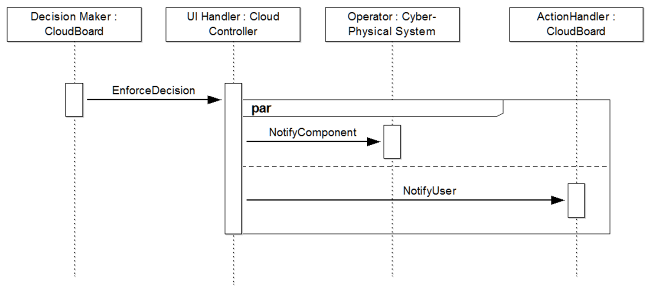
Decision Maker : CloudBoard
UI Handler : Cloud Controller
Operator : Cyber-Physical System
ActionHandler : CloudBoard
EnforceDecision
par
NotifyComponent
NotifyUser

Fig. 5. Information flows and component interaction for actualizing decisions.

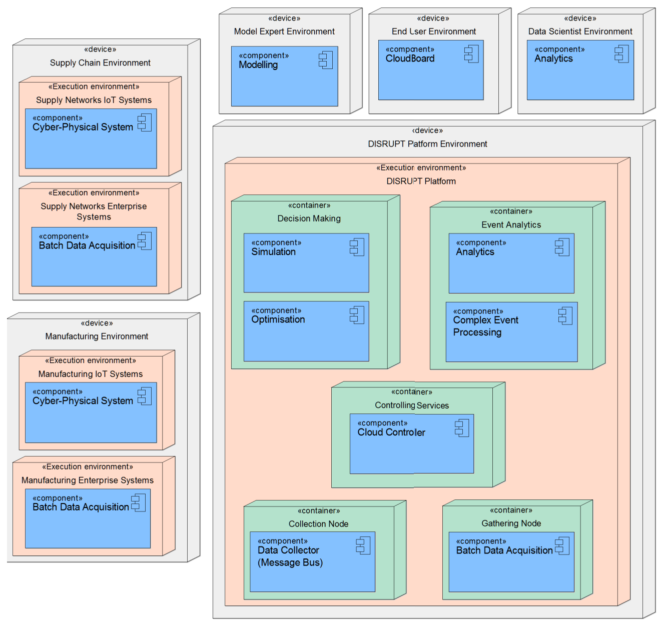
«device» Supply Chain Environment
«Execution environment» Supply Networks IoT Systems
«component» Cyber-Physical System
«Execution environment» Supply Networks Enterprise Systems
«component» Batch Data Acquisition
«device» Manufacturing Environment
«Execution environment» Manufacturing IoT Systems
«component» Cyber-Physical System
«Execution environment» Manufacturing Enterprise Systems
«component» Batch Data Acquisition
«device» Model Expert Environment
«component» Modelling
«device» End User Environment
«component» CloudBoard
«device» Data Scientist Environment
«component» Analytics
«device» DISRUPT Platform Environment
«Execution environment» DISRUPT Platform
«container» Decision Making
«component» Simulation
«component» Optimisation
«container» Event Analytics
«component» Analytics
«component» Complex Event Processing
«container» Controlling Services
«component» Cloud Controler
«container» Collection Node
«component» Data Collector (Message Bus)
«container» Gathering Node
«component» Batch Data Acquisition

Fig. 6. The DISRUPT physical architecture.

the Data Collection Framework services; (ii) the analysis of events both in terms of event detection and prediction; (iii) the decision making through simulation and/or optimisation; and, (iv) the controlling services, which host the operations performed by the Cloud Controller. The use of the container concept in this deployment diagram has the meaning of a distinct virtualisation object, which can be deployed in a distributed manner.

V. Concluding Remarks

This paper presents an architecture specification of an integrated system for managing decision making in manufacturing production and scheduling through the efficient identification and handling of disruptive events in the manufacturing ecosystem. Building upon established software architecture practices as well as emerging trends documented in Reference Architectures for smart manufacturing, the paper describes a modular, component-based system architecture using a viewpoint-oriented approach. The proposed architecture details the elements pertaining to the IIRA functional viewpoint from a software development perspective. External access to those elements functionality may be through (external) interfaces organised into RAMI 4.0 functional layers / services [33].

Targeting data-driven decision making, the proposed architecture viewpoints illustrate: (a) the role played by technologies that facilitate decision making and their interdependencies; (b) the flow of information across both vertical and horizontal dimensions; and, (c) the run-time structures of the system.

Our future work will focus on the development of the DISRUPT system as well as demonstrator tools for evaluating the proposed architecture in the two business cases.

ACKNOWLEDGMENT

This work has been supported by the EU H2020-FOF-11-2016 project DISRUPT (grant no 723541). The authors would like to thank all project partners for fruitful discussions.

REFERENCES

[1] T. Stock and G. Seliger, “Opportunities of sustainable manufacturing in industry 4.0,” *Procedia CIRP*, vol. 40, pp. 536 – 541, 2016. 13th Global Conference on Sustainable Manufacturing Decoupling Growth from Resource Use.

[2] L. D. Xu, E. L. Xu, and L. Li, “Industry 4.0: state of the art and future trends,” *International Journal of Production Research*, vol. 56, no. 8, pp. 2941–2962, 2018.

[3] Y. Lu, “Industry 4.0: A survey on technologies, applications and open research issues,” *Journal of Industrial Information Integration*, vol. 6, pp. 1 – 10, 2017.

[4] N. Jazdi, “Cyber physical systems in the context of industry 4.0,” in *2014 IEEE International Conference on Automation, Quality and Testing, Robotics*, pp. 1–4, May 2014.

[5] F. Shrouf, J. Ordieres, and G. Miragliotta, “Smart factories in industry 4.0: A review of the concept and of energy management approached in production based on the internet of things paradigm,” in *2014 IEEE International Conference on Industrial Engineering and Engineering Management*, pp. 697–701, Dec 2014.

[6] O. Givehchi, H. Trsek, and J. Jasperneite, “Cloud computing for industrial automation systems — a comprehensive overview,” in *2013 IEEE 18th Conference on Emerging Technologies Factory Automation (ETFA)*, pp. 1–4, Sept 2013.

[7] P. O’Donovan, K. Leahy, K. Bruton, and D. T. J. O’Sullivan, “Big data in manufacturing: a systematic mapping study,” *Journal of Big Data*, vol. 2, p. 20, Sep 2015.

[8] H. S. Kang, J. Y. Lee, S. Choi, H. Kim, J. H. Park, J. Y. Son, B. H. Kim, and S. D. Noh, “Smart manufacturing: Past research, present findings, and future directions,” *International Journal of Precision Engineering and Manufacturing-Green Technology*, vol. 3, pp. 111–128, Jan 2016.

[9] R. Y. Zhong, X. Xu, E. Klotz, and S. T. Newman, “Intelligent Manufacturing in the Context of Industry 4.0: A Review,” *Engineering*, vol. 3, no. 5, pp. 616–630, 2017.

[10] F. Oquendo, “Formally describing the software architecture of systems-of-systems with SosADL,” in *2016 11th System of Systems Engineering Conference (SoSE)*, pp. 1–6, June 2016.

[11] M. Guessi, V. V. G. Neto, T. Bianchi, K. R. Felizardo, F. Oquendo, and E. Y. Nakagawa, “A systematic literature review on the description of software architectures for systems of systems,” in *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, SAC ’15, (New York, NY, USA), pp. 1433–1440, ACM, 2015.

[12] P. Clements, F. Bachmann, L. Bass, D. Garlan, J. Ivers, R. Little, P. Merson, R. Nord, and J. Stafford, *Documenting Software Architectures: Views and Beyond, 2nd Edition*. Addison-Wesley Professional, 2011.

[13] P. Eirinakis, J. Buenabad-Chavez, R. Fornasiero, H. Gokmen, J.-E. Mascolo, I. Mourtos, S. Spieckermann, V. Tountopoulos, F. Werner, and R. Woitsch, “A proposal of decentralised architecture for optimised operations in manufacturing ecosystem collaboration,” in *Collaboration in a Data-Rich World* (L. M. Camarinha-Matos, H. Afsarmanesh, and R. Fornasiero, eds.), pp. 128–137, Springer International Publishing, 2017.

[14] E. Kavakli, J. Buenabad-Chávez, V. Tountopoulos, P. Loucopoulos, and R. Sakellariou, “An architecture for disruption management in smart manufacturing,” in *4th IEEE International Conference on Smart Computing (SMARTCOMP’18)*, pp. 279–281, 2018.

[15] F. Werner and R. Woitsch, “Data processing in industrie 4.0,” *Datenbank-Spektrum*, vol. 18, pp. 15–25, Mar 2018.

[16] P. B. Kruchten, “The 4+1 view model of architecture,” *IEEE Software*, vol. 12, pp. 42–50, Nov 1995.

[17] ISO/IEC/IEEE 42010:2011, “Systems and software engineering — Architecture description.”

[18] V. Kavakli and P. Loucopoulos, “Goal-driven business process analysis application in electricity deregulation,” *Information Systems*, vol. 24, no. 3, pp. 187 – 207, 1999. 10th International Conference on Advanced Information Systems Engineering.

[19] U. Frank, “Multi-perspective enterprise modeling (MEMO) conceptual framework and modeling languages,” in *Proceedings of the 35th Annual Hawaii International Conference on System Sciences*, pp. 1258–1267, Jan 2002.

[20] The Open Group, “OGAF v9.1, an Open Group Standard.”

[21] A.-W. Scheer and M. Nüttgens, *ARIS Architecture and Reference Models for Business Process Management*, pp. 376–389. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000.

[22] K. Kosanke, F. Vernadat, and M. Zelm, “Cimosa: enterprise engineering and integration,” *Computers in Industry*, vol. 40, no. 2, pp. 83 – 97, 1999.

[23] D. F. dos Reis Alves, R. de Campos, and F. B. de Souza, “Geram: building sustainable enterprises,” *IFAC Proceedings Volumes*, vol. 46, no. 24, pp. 602 – 607, 2013. 6th IFAC Conference on Management and Control of Production and Logistics.

[24] P. Hehenberger, B. Vogel-Heuser, D. Bradley, B. Eynard, T. Tomiyama, and S. Achiche, “Design, modelling, simulation and integration of cyber physical systems: Methods and applications,” *Computers in Industry*, vol. 82, pp. 273 – 289, 2016.

[25] Object Management Group, “Systems modeling language specification,” 2009.

[26] Industrial Internet Consortium, “The Industrial Internet of Things Reference Architecture,” Technical Report, Industrial Internet Consortium, Jan. 2017.

[27] ISO/IEC CD 30141:2016(E), “Information technology Internet of Things Reference Architecture (IoT RA),” 2016.

[28] IEC PAS 63088:2017, “Smart manufacturing - Reference architecture model industry 4.0 (RAMI4.0),” 2017. publicly available specification (pre-standard).

[29] A. Bousdekis, N. Papageorgiou, B. Magoutas, D. Apostolou, and G. Mentzas, “A real-time architecture for proactive decision making in manufacturing enterprises,” in *On the Move to Meaningful Internet Systems: OTM 2015 Workshops* (I. Ciuciu, H. Panetto, C. Debruyne, A. Aubry, P. Bollen, R. Valencia-García, A. Mishra, A. Fensel, and F. Ferri, eds.), pp. 137–146, Springer International Publishing, 2015.

[30] J. Lee, B. Bagheri, and H.-A. Kao, “A cyber-physical systems architecture for industry 4.0-based manufacturing systems,” *Manufacturing Letters*, vol. 3, pp. 18 – 23, 2015.

[31] M. P. Papazoglou, W. J. van den Heuvel, and J. E. Mascolo, “A reference architecture and knowledge-based structures for smart manufacturing networks,” *IEEE Software*, vol. 32, pp. 61–69, May 2015.

[32] A. Theorin, K. Bengtsson, J. Provost, M. Lieder, C. Johnsson, T. Lundholm, and B. Lennartson, “An event-driven manufacturing information system architecture for Industry 4.0,” *International Journal of Production Research*, vol. 55, no. 5, pp. 1297–1311, 2017.

[33] T. Bangemann, C. Bauer, H. Bedenbender, *et al.*, “Industrie 4.0 service architecture basic concepts for interoperability,” Technical Report, VDI/VDE Society Measurement and Automatic Control (GMA), status report, Nov. 2016.