2018 IEEE International Conference on Smart Computing

WiP: An Architecture for Disruption Management in Smart Manufacturing

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*—This paper reports the work in progress towards the specification of a conceptual architecture of a smart system for supporting the management of disruptions in the manufacturing domain. In particular, it proposes an approach to the description of the system architecture based on a number of interrelated viewpoints following the pertinent ISO 42010 standard. The approach is being developed in the context of the EU-funded H2020 DISRUPT project aiming to deliver a comprehensive data-driven solution for automated vertical and horizontal integration facilitating the transition into smart manufacturing.**

***Index Terms*—Industry 4.0, smart manufacturing, architecture description**

I. Introduction

Smart Manufacturing describes the convergence of the digital and physical worlds in the manufacturing domain. The aim is to achieve seamless integration and cooperation among products, equipment, humans, and organizations, thus enhancing efficiency and agility of manufacturing processes [1]. The increasing complexity and dynamics of processes in smart manufacturing, leads to a high vulnerability to disturbances during production processes. Furthermore, due to the extensive connectivity between devices and processes in the smart manufacturing model (both within the factory and its ecosystem), a system failure may lead to larger losses from the ripple effect. Losses could mount not only from the affected system but also the other dependent businesses in the supply and distribution chains. As a consequence, managing disruption in smart manufacturing is becoming an important challenge [2].

Disruption management aims at coping with unexpected events to mitigate the effect of disruption in real-time [3]. The disruption management lifecycle includes the following phases: collection, detection and analysis of disrupting events, decision making in order to develop countermeasures to solve the problem induced by the disrupting event and reaction to implement the solution defined in the previous phase [2]. In the context of smart manufacturing, disruption management deploys a number of technologies, such as, data collection enabled through Internet-of-Things (IoT) technologies, Cyber-Physical systems (CPS) used for monitoring and control of physical manufacturing resources, decision support driven by data analytics and complex event processing [4], while cloud environments may be used for computational or storage support. Such technologies have been presented in the literature and have been the subject of various projects [5], [6] focusing mostly on production systems, whilst less attention has been given in other application areas such as logistics systems [7].

This short paper reports on the initial specification of a software system architecture to support the management of disruptions in smart manufacturing that takes a wider perspective correlating the production process with information derived from the whole value chain. This is ongoing work carried out in the context of the EU-funded H2020 DISRUPT project[1].

The paper is structured as follows. Section II, presents the methodology used for specifying the system architecture. The detailed description of the proposed architecture is provided in Section III. Finally, Section IV concludes and provides pointers to future developments.

II. Viewpoint-oriented Architecture Specification

The notion of viewpoint-oriented architecture in requirements and software engineering originates in the 1990s as a way to address the complexity of software development in a setting with many actors, using various representation schemes, having diverse domain knowledge and different development strategies [8], [9]. As a result, several architecture frameworks have been proposed, which are essentially viewpoint classification schemes. The classification by Kruchten [10], proposed in 1995, is probably the best known classification and is still widely used. These early ideas on viewpoint-oriented software engineering have found their way into the ISO/IEC 42010:2007 standard, revised in 2011, in which a viewpoint is defined as "a way of looking at systems" [11]. According to the above, viewpoints are stakeholder-centric; their content is determined by their relevance to a stakeholder's concerns. In addition, each viewpoint prescribes one/set of language(s), modelling technique(s), or analytical method(s) (model kinds) which, when applied to a particular system of interest, result in a specific system description (view).

Obtaining a viewpoint-oriented architecture specification involves the following steps: (a) identification of the stakeholders having concerns considered fundamental to the architecture

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

of the system; (b) identification of the architecture viewpoints, providing a name for each viewpoint, a listing of architecture-relevant concerns to be framed by this viewpoint, and a listing of the typical system stakeholders; and (c) identification of the model kind(s) used in each viewpoint.

In the context of enterprise systems, viewpoints are prominently present in the Open Group TOGAF architectural framework [12]. Similarly, industrial IoT systems (in Energy, Healthcare, Manufacturing, Public Domain and Transportation domains) are also described through viewpoints (business, usage, functional and implementation viewpoints) by the Industrial Internet Reference Architecture (IIRA) [13]. For manufacturing, IIRA is complemented by the Reference Architecture Model Industrie 4.0 (RAMI4.0) [14] through a cubic, 3-viewpoint model that allows assets to be (a) described in the form of functional layers, (b) tracked over their entire lifetime, and (c) assigned to technical and/or organisational hierarchies [14]. Assets include products, machines, production lines, software: anything that is of value to a company. The proposed architecture details the elements pertaining to the IIRA functional viewpoint from a software development perspective. External access to those elements' functionality will be through (external) interfaces organised into RAMI4.0 functional layers.

In line with the above, the next section describes an initial architecture description as system-specific views (models) that conform to three complementary viewpoints.

III. A Disruption Management System Architecture

The architecture vision of the system is guided by the desire to support knowledge-driven decision making in the production and scheduling through the efficient identification and handling of events in the manufacturing ecosystem that could disrupt its operations. The system stakeholders include system users (domain experts, operational managers and decision makers), system administrators, as well as technology providers (including the suppliers, the developers/integrators, the testers and the maintainers of the system).

The following list provides an overview of the concerns (to be further elaborated in each viewpoint) that are considered fundamental to the architecture of the system:

- The purpose of the system is to support the disruption management lifecycle.
- The system will be added onto an existing factory ecosystem.
- The system will be based on a distributed architecture, consisting of loose coupling of (potentially existing) functional components, to enable flexibility in the system components and their functionality. This should reduce the risk for a vendor lock-in.

Based on the above the proposed architecture specification focuses on the identification of the system components and the interactions between them in order to achieve the intended system functionality taking into consideration non-functional issues and not how each individual component works.

A. Architecture Specification

Three architectural viewpoints have been considered, in accordance to the guidelines proposed in [15], whereby the system is seen as: a set of implementation units (logical view); a set of runtime elements interacting to carry out the systems work (informational view); and a set of elements existing in and relating to external structures in its environment (physical view). An overview of each viewpoint is shown in Table I.

TABLE I
The logical, informational and physical viewpoints

The Logical Viewpoint	
VP Overview	Describes the system main components, their functionality and interfaces
Concerns	Functional capabilities that support the functional requirements, flexibility to support functional changes, external interfaces, and relationships between system components
Typical stakeholders	All stakeholders
Model kinds	UML class diagrams, UML component diagrams, or both
The Informational Viewpoint	
VP Overview	Describes a complete but high-level view of run-time information flow between system components
Concerns	Run time information flow between system components; real-time requirements, timeliness, latency, and age; references and mappings; transaction management and recovery; data quality; data volumes; archives and data retention; and regulation.
Typical stakeholders	Primarily users, acquirers, developers, and maintainers, but most stakeholders have some level of interest
Model kinds	UML information flow diagrams, UML sequence diagrams or both
The Physical Viewpoint	
VP Overview	Describes the environment into which the system will be deployed, including dependencies it has on the environment, and the mapping of system components to the environment
Concerns	Types of hardware required, network requirements and physical constraints
Typical stakeholders	System administrators, developers, testers
Model kinds	UML deployment diagrams

B. Architecture Description

Fig. 1 demonstrates the development of a system-specific view that conforms to the logical viewpoint using the UML component diagram formalism. The figure presents the system components, their relations and their internal interfaces. These include: the Cyber-Physical System (CPS) component, which continuously collects and processes data from the plant floor and the supply chain and enacts high-level decisions upon the plant floor through IoT; the Data Collection Framework (DCF) that collects, processes and stores data from enterprise information systems and stores processed data from CPS; the Complex Event Processing component which identifies (from analysing data in the DCF) events beyond normal operation whose impact is assessed by the Simulation component; the Data Analytics component that seeks to identify (from data in DCF) abnormal trends that may lead to disruptions; the Modelling component which provides production models that are optimised by the Optimisation component and evaluated based on Simulation; the CloudBoard component which is the user interface to access all system functions; and the Cloud

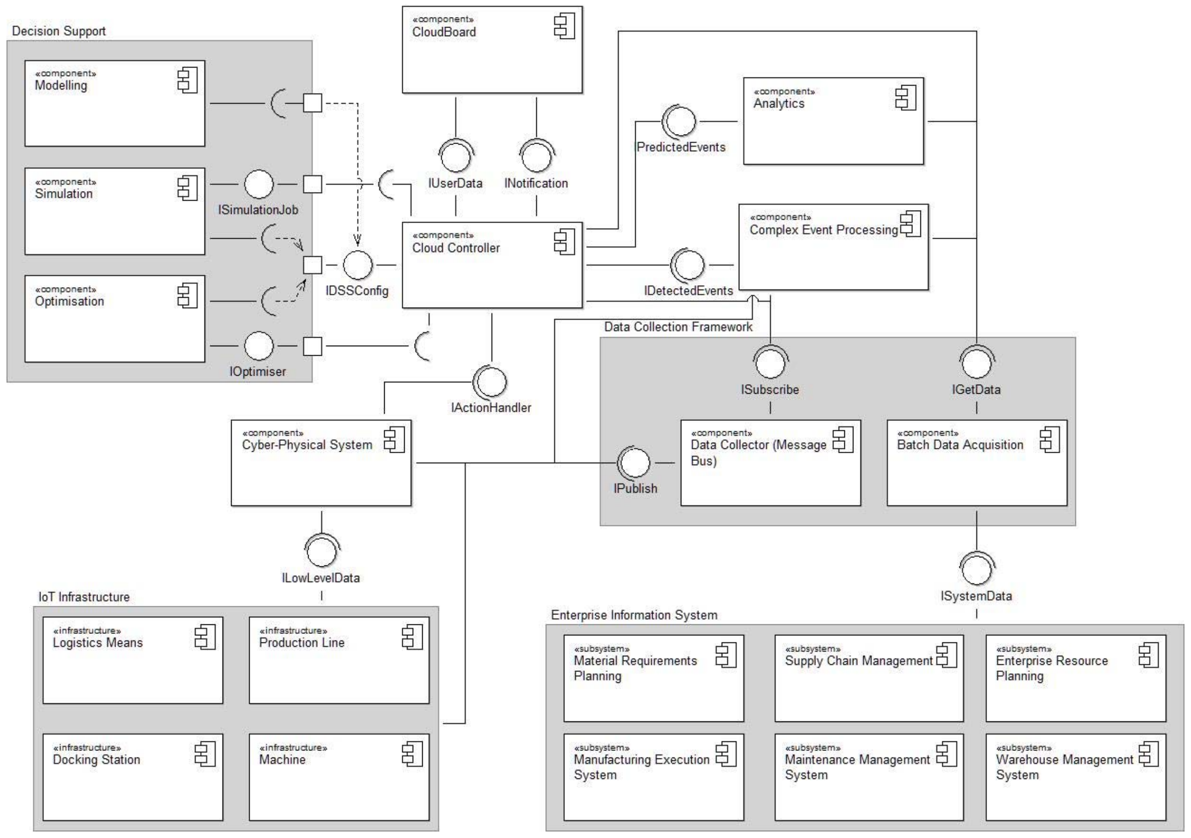
Decision Support
«component» Modelling
«component» Simulation
«component» Optimisation
ISimulationJob
IDSSConfig
IOptimiser
«component» CloudBoard
IUserData
INotification
«component» Cloud Controller
«component» Analytics
PredictedEvents
«component» Complex Event Processing
IDetectedEvents
IActionHandler
«component» Cyber-Physical System
Data Collection Framework
ISubscribe
IGetData
IPublish
«component» Data Collector (Message Bus)
«component» Batch Data Acquisition
ILowLevelData
ISystemData
IoT Infrastructure
«infrastructure» Logistics Means
«infrastructure» Production Line
«infrastructure» Docking Station
«infrastructure» Machine
Enterprise Information System
«subsystem» Material Requirements Planning
«subsystem» Supply Chain Management
«subsystem» Enterprise Resource Planning
«subsystem» Manufacturing Execution System
«subsystem» Maintenance Management System
«subsystem» Warehouse Management System

Fig. 1. Component diagram of the system architecture.

Controller component which enacts user requests (received through the CloudBoard) and facilitates the integration and interaction between the different components.

IV. Concluding Remarks

This short paper presents work in progress towards the architecture specification of an integrated system for managing disruptions in smart manufacturing following a viewpoint-oriented paradigm. Future work will specify the viewpoints in further detail and elaborate the interactions between components of the architecture. In addition, appropriate demonstrator tools will be developed to evaluate the proposed architecture.

Acknowledgment

This work has been supported in the context of 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] Smart Manufacturing Leadership Coalition, https://smartmanufacturingcoalition.org/

[2] N. Galaske and R. Anderl, "Disruption Management for Resilient Processes in Cyber-Physical Production Systems", *Procedia CIRP* 50, 442-447, 2016.

[3] G. Yu and X. Qi, *Disruption Management*, World Scientific Publishing, Singapore, 2004.

[4] P. Zheng, H. Wang, Z. Sang, et al, "Smart manufacturing systems for Industry 4.0: Conceptual framework, scenarios, and future perspectives", *Frontiers of Mechanical Engineering*, 13(2), 137-150, 2018.

[5] Y. Liao, F. Deschamps, E. de Freitas Rocha Loures and L. F. Pierin Ramos, "Past, present and future of Industry 4.0 – a systematic literature review and research agenda proposal", *International Journal of Production Research*, 55(12), 3609-3629, 2017.

[6] P. Leita, A. W. Colombo, S. Karnouskos, "Industrial automation based on cyber-physical systems technologies: Prototype implementations and challenges", *Computers in Industry*, 81, 11-25, 2016.

[7] E. Hofmann and M. Rüsch, "Industry 4.0 and the current status as well as future prospects on logistics", *Computers in Industry*, 89, 23-34, 2017.

[8] A. Finkelstein, J. Kramer, B. Nuseibeh, et al., "Viewpoints: A Framework for Integrating Multiple Perspectives in System Development", *International Journal on Software Engineering and Knowledge Engineering*, 2(1), 31-58, 1992.

[9] G. Kotonya and I. Sommerville, "Viewpoints for Requirements Definition", *IEE/BCS Software Engineering Journal*, 7(6), 375-387, 1992.

[10] P. B. Kruchten, "The 4+1 View Model of Architecture", *IEEE Software*, 12(6), 42-50, 1995.

[11] ISO/IEC/IEEE 42010:2011, "Systems and Software Engineering – Architecture Description", 2011.

[12] The Open Group, TOGAF v9.1, an Open Group Standard, available at http://pubs.opengroup.org/architecture/togaf9-doc/arch

[13] Industrial Internet Consortium, "IIRA: The Industrial Internet of Things Reference Architecture", Tech. report, 2017.

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

[15] P. Clements, F. Bachmann, L. Bass, et al, "A Practical Method for Documenting Software Architectures", working paper, 2002. Available from http://repository.cmu.edu/compsci/671