RESTful Cyber-Physical Systems toward Industrial Automation

Semester project, Jonas Brütsch
External supervisors: Prof. Simon Mayer, Iori Mizutani

September 2020

Motivation

Automation of processes has been a core challenge for modern industry to relieve humans from repetitive or dangerous tasks. Machines can perform these tasks with higher precision, faster, more efficient and at lower cost, but the design and operation of automated systems require a significant share of domain and expert knowledge [4]. In a manufacturing automation system, the heterogeneity in the machines (e.g,, data formats, protocols, and communication interface) makes the automation system increasingly complex. To increase uniformity and consistency of automation system management, many modern factories are built based on a hierarchical model as often referred to as "Automation Pyramid" (standardized as IEC 62264 [2], see Figure 1). Some layers share the similar functionalities (e.g., SCADA and PLC can be merged into one "Control" layer), or some factories don't even implement a corresponding layer because it is often deemed economically infeasible as only a fraction of the functionality is used (e.g., the size of the factory is to small and SCADA doesn't bring financial benefits to optimize the operation). This modularization and the heterogeneity problem make it difficult to provide necessary explanations about the automation systems on higher layers. In order to solve this challenge, among others, more research in the field of *Explainability* must be done.

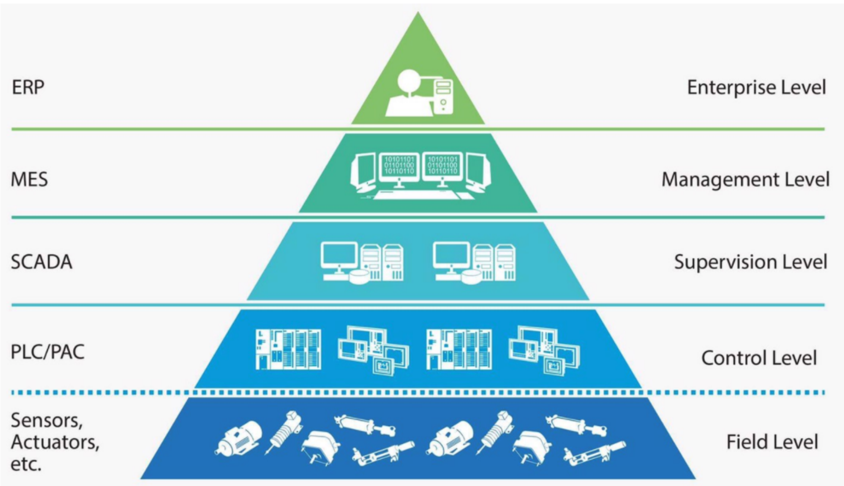

Figure 1: Automation Pyramid[1]

Explainability is one of the emerging requirements in the automation field to verify complex dependencies in the system or the executing tasks by providing "explanations" for humans or machines [3]. However, the definition of explanation diverges in many ways; i.e., there are many open research questions to address what makes system explainable. To take an example, the provenance information enables traceability of products being manufactured in a manufacturing system (e.g., identifiers, locations, status in the production processes), but the implementation of a traceability service application is entirely different from the one that provides other information, such as process logic management (e.g., fault detection and diagnosis of hardware and software components). In order to contextualize the approach to realize explainable systems, we will first build a Proof of Concept automation system with the REST architectural style [5] for service level abstraction, and investigate what is to be explained, how it should be delivered, and specifically, why it is difficult to achieve an explainable system.

[1] M.I.A.C. Automation Co.,Ltd.: `http://www.miac-automation.com/mes-oee-track-and-trace/`

Representational state transfer (REST) is a resource-oriented architectural style designed for distributed applications, such as the Web. The REST principles propose the use of simple, uniform interfaces to interact with resources by the transfer of their representations, assuming that the representations may refer to virtual as well as to physical objects. We apply the same principles to the CPS by treating physical components in the systems in the same way as the other resources on the Web – creating Web of Things environments which seamlessly integrate other services on the Web.

On the other hand, OPC Unified Architecture (OPC UA) [1] is an industrial standardization effort to tackle this heterogeneity in automation engineering. OPC UA spans across many different specification documents that describe services, address space, information model, communication protocol, and so on – the architecture is developed to accomplish multiple design specifications: functional equivalence, platform independence, secure, extensible, and comprehensive information modeling [2]. The OPC UA enables us to anticipate a paradigm shift from the Automation Pyramid model to the RESTful mesh network model as depicted in Figure 3. The semantic integration layer provides service-orientated semantic queries for different explainability applications, and different semantic integration layers can be implemented adjacent to the actual Automation Pyramid.

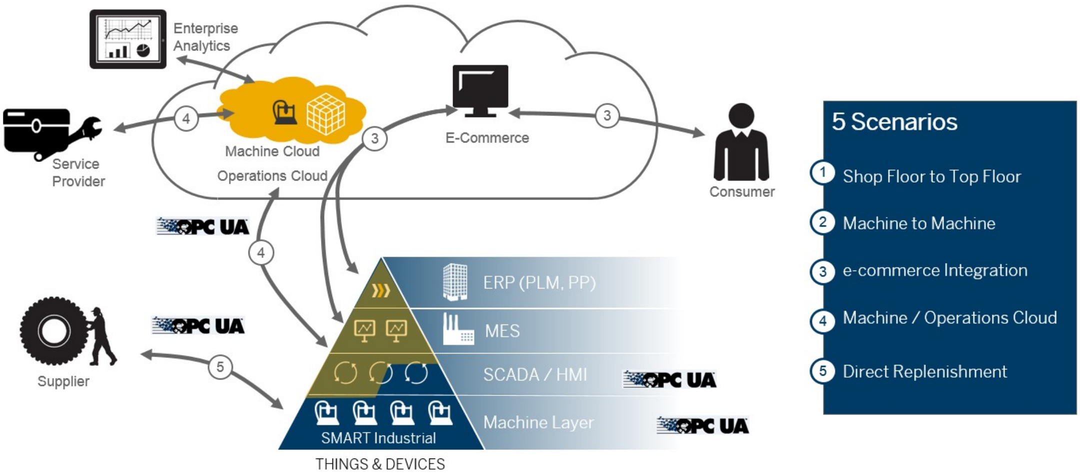

Figure 2: OPC UA extension to Automation Pyramid[3]

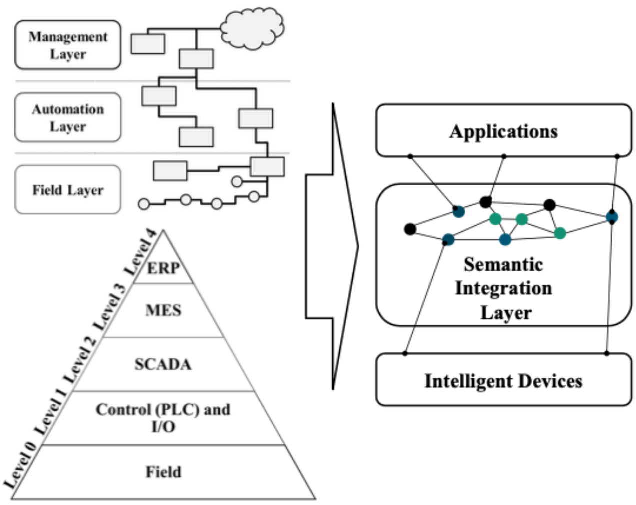

Figure 3: Paradigm shift from the Automation Pyramid to the abstraction with Semantic Integration Layer [6]

[2]https://opcfoundation.org/about/opc-technologies/opc-ua/
[3]SAP SE and OPC Foundation: https://opcconnect.opcfoundation.org/2016/03/reshape-the-automation-pyramid/

Project Description

In this project, the student will develop a PoC manufacturing automation system with RESTful API for all the physical components in the Field, Control, and Supervision Levels while conceiving the adaptation of/extension to the OPC UA and the semantic integration layer. With the developed system, the feasibility and the trade-off between explainability and efficiency/responsiveness shall be further investigated. The structure of the Hypermedia Factory is shown in Figure 4. To simulate a real-world case, several robot arms (uArm [7]) are used in combination with different sensors and actuators.

Work Packages

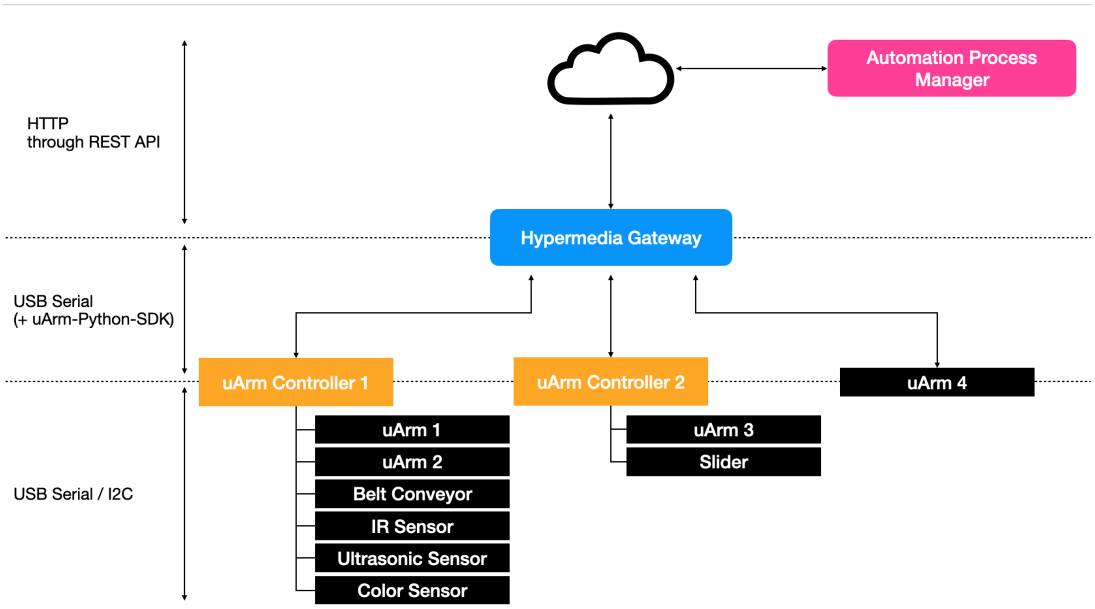

Figure 4: Overview of the Hypermedia Facroty

WP1: uArm controller firmware

Some hardware components can only be accessed through the MCU (ATmega 2560); i.e., the controller. The firmware on the controller exchanges serial messages with the host machine, and translate the command from the host machine for the field devices. To enable this connection between Hypermedia Gateway and controllers, a simple serial protocol has to be developed and implemented. In addition to the control messages, a feedback mechanism has to be implemented for some of the hardware components (e.g. uArm, Belt Conveyor).

WP2: Hypermedia Gateway

After finishing the firmware, the student builds the gateway to allow RESTful resource control/management for the field devices.

- Design of the API specification
- Selection of a programming language and framework to implement the API handler
- Adaptation of the serial protocol to communicate with the controllers from WP1
- Validation of the implemented firmware from WP1 with multiple tests.

WP3: Automation of the processes

The student creates an event flow manager to coordinate the devices to operate a manufacturing process.

References

[1] International Electrotechnical Commission et al. Iec tr 62541-1: 2016-opc unified architecture-part 1: Overview and concepts. *IEC, Geneva, CH, Technical Report*, 2016.

[2] IEC IEC. 62264-1: Enterprise-control system integration–part 1: Models and terminology. *IEC: Geneva, Switzerland*, 2013.

[3] Maximilian A Köhl, Kevin Baum, Markus Langer, Daniel Oster, Timo Speith, and Dimitri Bohlender. Explainability as a non-functional requirement. In *2019 IEEE 27th International Requirements Engineering Conference (RE)*, pages 363–368. IEEE, 2019.

[4] Thilo Sauter, Stefan Soucek, Wolfgang Kastner, and Dietmar Dietrich. The evolution of factory and building automation. *IEEE Industrial Electronics Magazine*, 5(3):35–48, 2011.

[5] Rainer Schiekofer, Andreas Scholz, and Michael Weyrich. Rest based opc ua for the iiot. In *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, volume 1, pages 274–281. IEEE, 2018.

[6] Georg Ferdinand Schneider. *Semantic Modelling of Control Logic in Automation Systems - Knowledge-Based Support of the Engineering and Operation of Control Logic in Building and Industrial Automation Systems*. PhD thesis, Karlsruher Institut für Technologie (KIT), 2019.

[7] Ufactory. Web store: uarm. `https://store.ufactory.cc/pages/uarm`, 2020. [Online], (accessed 18.09.2020).