A rapid control prototyping platform methodology for decentralized automation

Florian Kästner, Benedikt Janßen, Sebastian Schwanewilms, Michael Hübner
Chair for Embedded Systems for Information Technology
Ruhr-University Bochum
Email: {Florian.Kaestner, Benedikt.Janssen; Sebastian.Schwanewilms; Michael.Huebner}@rub.de

***Abstract*— Today's industrial requirements regarding the ability of embedded devices used for decentralized automation are increasing. Industrial providers of automation equipment strive to make their products and thus, industrial plants, smarter to raise efficiency. This evolution is based on new technologies like machine learning, predictive maintenance, sensor fusion and advanced process controls. These techniques require performance and energy efficient hardware platforms supporting a fast execution of computational intensive algorithms in compliance with real-time constraints. Therefore, to achieve these targets in a cost-efficient manner, the sharing of hardware resources to implement advanced process controls or machine learning algorithms is beneficial. Further, if different institutions integrating intellectual property (IP) into a single platform a certain degree of isolation is mandatory to protect their IP against theft or manipulation. In this paper, we propose a rapid control prototyping platform supporting the sharing of resources in an isolated manner to evaluate new control or monitoring strategies on a single platform with the help of Linux Containers for process isolation, MQTT for interprocess communication, OPC UA for vertical integration and partial bitstreams.**

Keywords*—*Rapid Control Prototyping; MQTT; LXC;OPC UA; process isolation; Advanced Process Controls

I. Introduction

Decentralized automation is one of the key technologies within the megatrends Industry 4.0 and Internet of Things (IoT). Replacing expensive and not optimized central programmable logic controllers (PLC) with distributed smart embedded devices offers many optimization opportunities like improving power, cost or resource-efficiency. Further, performance capabilities of microprocessors and System on Chips (SoCs) are increasing while costs are decreasing. Meanwhile, the traditional automation hierarchy is flattened as optimized hardware offer the opportunity to shift many tasks from the higher levels of the hierarchy to one level which is called Cyber Physical Production System (CPPS) and summarizes the distributed intelligence of the embedded platforms. While traditional embedded systems are developed as standalone devices, the key concept of CPS is the interaction of several devices to form distributed intelligence. For further investigations, we refer to the following definition of CPS [1].

Fig.1 shows the traditional automation pyramid on the left side, consisting of 5 different levels where the most lower level, called field level, is the spatial closest one to the process. The higher the level the closer the tasks are related to the Information Technology (IT) area and thus more distant to the automation area.

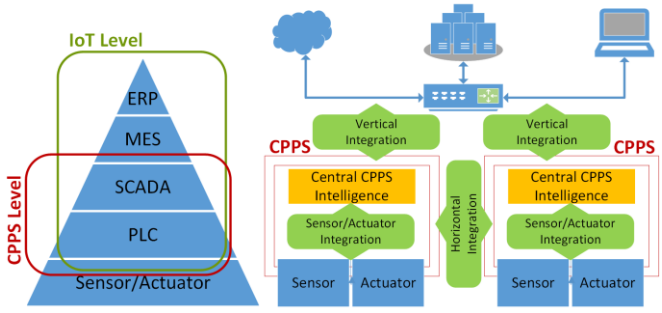

Fig. 1. Automation system hierarchical pyramid

Traditionally, the separation line between those areas is placed between the control-level (PLC) and the process-management-level (SCADA) as can be seen in Fig.1. The higher levels, plant-management- (MES) and company-level (ERP) are belong to the IT area. Due to the capabilities of modern embedded platforms, such as the Zynq 7000 SoC [2], consisting of a dual-core ARM A9 and a Field Programmable Gate Array (FPGA), the amount of higher level tasks such as Machine Learning (ML), Advanced Process Control (APC) or sensor fusion (SF) which are able to implement on such a platform effectively is increasing. This fact leads to a natural evolution to flatten this hierarchy and represents the starting point developing a decentralized automation in a CPPS environment. In this case sharing resources is beneficial, because industrial providers of automation technology are able to include their knowledge about their products and the process with the help of algorithms like ML, SF or APC. Therefore, useful sensor-actuator combinations have to be found. Further, a rapid prototyping platform has to be built allowing different institutes to integrate and their algorithms, using common interfaces to share results gained by the algorithms and evaluate the interactions between those in a secure manner.

II. Rapid Control Prototyping Methodology

If different providers of knowledge are integrating their Intellectual Property (IP) on a single platform we have to ensure a certain degree of security to protect the individual IP. Further, a simple and generic interprocess communication has to be developed. In addition, if the providers of IPs want to evaluate hardware accelerators we have to provide an opportunity to integrate and interface the hardware implementations in a simple and secure way. However, the first step is to choose the components and the way of integration.

A. *Target Platform and Integration*

Decentralized automation in a CPPS environment focuses on distributed intelligence and subsystems without the need of a PLC with higher level tasks. Therefore, a powerful and scalable embedded platform is needed. For the evaluation, we chose the Zedboard equipped with a Zynq-7000 SoC [2], with a good tradeoff between performance, power consumption and cost. The reconfigurable logic on the Zynq can be used to implement and evaluate hardware accelerated algorithms or fieldbuses. While facing the problem of process isolation and communication, we propose a hardened embedded Linux operating system implemented with Yocto [5]. Another important aspect designing a methodology for a rapid control prototyping platform is the vertical, horizontal and sensor/actuator integration, as depicted in Fig.1. While the horizontal and sensor/actuator communication is real-time driven and heavily application depended, the vertical integration is service-oriented to gain useful information to the management level. For this type of communication OPC Unified Architecture (UA) [3] is established as a widely used protocol based on TCP/IP. OPC UA is an open protocol with a scalable stack using a client-server model with a proven security technology based on Transport Layer Security. The user is able to choose among different data models depending on his needs and application preconditions. We propose to equip the rapid control prototyping platform with a simple OPC UA server, based on the open62541 [4] Stack, sharing the data model. Thus, different vendors can monitor the process remotely depending on their authorization.

B. *Process Isolation and Interprocess Communication*

Process Isolation is an important aspect if different vendors share a common hardware platform. For this purpose, we propose a virtualization method called Linux Containers (LXC). This allows running multiple isolated, unprivileged applications using a single Linux kernel. The main functionalities used for this virtualization method are Linux' cgroups and namespaces isolation, which offer the possibility to configure the container depending on their needs and authorization. This allows us to mount only specific file trees into the file system of the container, defining permissions, limit or prioritize resources or adjust networking properties of the container, which forms the base system to implement the application of a specific vendor. For this purpose, LXC can be configured with a wide range of application Programming interfaces (APIs). We further propose the Message Queue Telemetry Transport (MQTT) [6] as an interprocess communication. MQTT is based on the TCP/IP protocol and is using a publisher/subscriber model. Therefore, for each application a MQTT Client is running in a separate thread sharing a protected common data model.

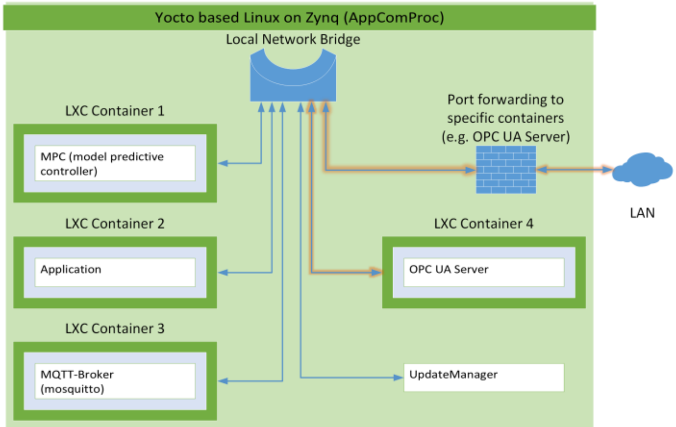

Fig. 2. Automation system hierarchical pyramid

The topic list is the common developing supply source between the different vendors. Thus, they can decide which data they would like to receive or supply to other vendors or peripherals. The broker, who organizes this communication, runs in a separate container. Fig.2 shows the system view of the platform. Further, this configuration is used to configure custom recipes configuring a hardened Linux OS distribution.

C. *Hardware Usage and Interfacing*

The opportunity to accelerate tasks in hardware is beneficial in terms of meeting real-time constraints of complex algorithms. Therefore, we propose to offer the vendors fixed areas on the FPGA with fixed interfaces controlled by the processing system. Within these P-blocks vendors have the opportunity to evaluate hardware implementations and interface them through the generic Userspace I/O (UIO). The access to these devices are later linked to the container.

III. Conclusion

In this paper, we have proposed a design methodology to build a custom and effective rapid control prototyping platform in the decentralized automation area, which allows different vendors to integrate and evaluate their IP as software or hardware implementations on a single platform in secure and simplified manner.

Acknowledgement

This work was done under the support of BMBF project DnSPro (16ES0392).

References

[1] E. A. Lee, "Cyber Physical Systems: Design Challenges," ISORC 2008, Orlando, FL, 2008, pp. 363-369l

[2] Xilinx, "Zynq-7000 All Programmable SoC", Technical Reference Manual, Xilinx, 2014

[3] *OPC Unified Architecture*, 2008, [online], Available: http://www.opcfoundation.org/,

[4] *open62541*, 2015, [online], Available: https://open62541.org/

[5] Yocto Project, 2010, [online], Available: https://www.yoctoproject.org/

[6] MQTT Protocol, 2013, [online], Available: http://mqtt.org