# PlugBot Architecture for Modular Manufacturing

Georg Weichhart[1], Markus Ikeda[1], Matthias Propst[1] and Roman Froschauer[2]

[1] PROFACTOR GmbH, Im Stadtgut D1, 4407 Steyr, Austria
[2] University of Applied Sciences Upper Austria, Wels, Austria

**Abstract**

PlugBot was an Austrian research project (2019-2021) in which an architecture to integrate heterogeneous (robotic) systems has been developed. Modeling system capabilities for task execution as atomic skills is a concept that allows semantic unification for orchestration of multiple skills along workflows as well as hardware abstraction and implementation agnostic workflow modelling without taking concrete parameters of the executing devices into account. The architecture and approach has been evaluated in multiple use cases in the manufacturing domain. Future work will include researching the applicability of the concept for mobile robots.

**Keywords**

Architecture for integration, enterprise interoperability, robotics, skill-based programming

## 1. Introduction

Enterprise Interoperability has its roots in research fields Enterprise Modelling, Enterprise Architecting and Enterprise Integration [1]. It is a model driven research domain, where multiple interacting systems are supported in their communication, coordination or collaboration [2]. These fields research the interaction of systems from an information systems point of view, where systems include human and artificial agents that interact [3]. To distinguish interoperability and integration, we can take a look at a continuum, where the interoperability end aims at a loose coupling and the other end (integration) at tight coupling of systems. A system is an integrated unit, in contrast to that, a system-of-systems is a loose coupled unit where the systems have a great degree of autonomy but still the interaction needs to be interoperable [4]. Systems-of-systems are modular systems which – from a theoretical point of view - are very flexible due to the autonomy of the systems. In applied production research, the systems-of-systems approach supports decentralized and modular manufacturing. Here recently cyber-physical systems (CPS) research has been developed as an important paradigm forming Cyber-Physical Production Systems (CPPS) [4]. In production in general, but more specifically in robotics, interoperability is becoming an important aspect that needs to address technical interoperability, semantic interoperability and *physical* interoperability. This latter part requires more attention as it is often overlooked in enterprise interoperability, discussing only information systems and flows of data / information / knowledge.

Within the PlugBot project an architecture and framework has been researched that enables modular manufacturing with special attention to robotics. In the following section, we discuss the state of the art, followed by a description of obtained research results and an example use cases, showing the potential of the solution. Finally, we discuss concluding remarks.

## 2. State of the art

Proceedings of the Workshop of I-ESA'22, March 23–24, 2022, Valencia, Spain
EMAIL: Georg.Weichhart@profactor.at (G. Weichhart); Markus.Ikeda@profactor.at (M. Ikeda); Matthias.Propst@profactor.at (M. Propst); roman.froschauer@fh-wels.at (R. Froschauer)
ORCID: 0000-0002-1405-5825 (G. Weichhart); 0000-0001-6754-5783 (M. Ikeda); 0000-0002-7724-9037 (M. Propst); 0000-0001-6952-285X (R. Froschauer)

In this section we first, present trends and the state of the art of modular manufacturing system, give special attention to skill-based robotic systems and then discus how modular systems can be made interoperable.

## 2.1. Challenges in modular cyber-physical production systems

A system is a (conceptual) unit that consists of modules. It has an interface to its environment and can therefore be distinguished from it. Through that interface a system interacts with the environment and other systems [5, 6]. A system has a purpose, or goal. To realize that purpose, the modules provide required functions. These modules do not work independently but are dependent on the system to which they provide the function. In contrast to that, a system has a certain level of independence.

A Cyber-Physical System (CPS) is a system, which consists of a physical and a cyber module, realizing the required functionality. The interfaces of such systems are sensors and actors that support interaction. In addition to this, the software often supports the networking of such systems and the communication and collaboration with other systems. Multiple of such systems form Cyber-Physical Production Systems in the manufacturing domain forming a System-of-Systems (SoS) [4].

In the manufacturing domain, also the shop floor may be addressed as an integrated system or a system-of-systems consisting of loose coupled systems. The difference lies in the structural modularity, which allows more adaptiveness and flexibility [7]. Of course, structural aspects can only provide the basic possibilities [8]. One way to implement a modular and dynamic system is given through the Multi Agent System (MAS) architecture [9, 10] .

A challenge to approaches that enable such modular and agile CPPS is the disintegration of the Automation Pyramid, which is the currently implemented architecture. This hierarchical organization of IT systems is based on the planning horizon implemented by the individual systems. In existing manufacturing systems it's hard to move towards more modular and agile manufacturing.

One of the challenges is, that the Automation Pyramid is often implemented as integrated system where every layer is strongly linked to the next and previous layer. In order to reach a new improved modular and flexible architecture the paradigms of loose coupling and interoperability are necessary.

## 2.2. Skill-based modular robotics

Skill-based modeling and operation of industrial software & automation systems including robots is widely researched as accepted method for enabling hardware agnostic programming and loosely coupling of automation components towards plug & produce in production plants [11-13] . The modeling of skills is mostly based on (1) the definition of machine capabilities as a set of specific properties and the ability to perform specific operations on a process resource and (2) the externally triggered execution of these operations as presented in [14-15] . By this means skill-based programming deals with modeling of a subsequent order of executing specific process functions (i.e. skills) without specific knowledge of its implementation on the machine or robot. Implementing these concepts several approaches of skill-based programming have been applied in the field of robotics [13, 16] where the functionality of a complex machine featuring several robots has been split into modules representing different capabilities (i.e. executable skills such as Grasp, Move, Grind, Drill, …) to allow for a more generic way of programming, not relying on specific communication interfaces between the modules (i.e. robot, gripper, machine). Today the demand of a generic communication method to call skills is increasingly fulfilled by the OPC UA standard. In respect to robotics, there is a companion specification available regarding the basic capabilities of the robot [17], as well as the well accepted idea of representing capabilities, and its corresponding skills, using the OPC UA information model, method calls or programs [18]. Furthermore, using OPC UA and its Local Discovery Services [19], devices can be automatically discovered within a local network. As presented in [20], the production process is operated by using recipes representing a sequence of skills to be executed by loosely coupled and modular devices. Another implementation for a modular robotic application using hardware from different vendors has also been presented in [21].

## 2.3. Enterprise interoperability in manufacturing

Enterprise Interoperability may be seen as one end of a continuum ranging from tight integration of systems to loose coupled systems [22]. The loose coupling is a challenge. From a systems design and engineering point of view, integrated systems engineering is very different to system-of-systems engineering [4, 23]. In interoperability, both properties have to be enabled, independency of self-contained systems that have a certain degree of autonomy, and seamless interaction where these systems communicate, coordinate and collaborate to reach a common, higher goal [24, 25].

Process models can be used to orchestrate independent engineered resources (including robots) [7, 26, 27]. These models help to increase interoperability on the organizational level. In addition to this, the technical and the semantic layer need to be addressed to cope with the interoperability challenge.

The level of independence between systems is dependent on the use of standards on technical level (e.g. OPC UA). Such standards provide clear interfaces between systems. The second level that must be considered is the semantic interoperability layer, where data semantics needs to be understood by all participants that use a certain interface [22].

## 3. PlugBot architecture for manufacturing systems-of-systems

PlugBot was an applied research project in Austria, running from 2019-2021. The overall goal was to provide tools for the development and integration of heterogeneous (robotic) automation systems. The targeted application domain is production and manufacturing. This section introduces the architecture and briefly discusses the two use cases that evaluate the functionality of the PlugBot approach.

## 3.1. Architecture

As explained in [14] "*for seamless interaction of high-level control units and low level devices on the network, logical and behavior level, a standardized interface and interaction concept is crucial*". This also applies to heterogenous robot systems consisting of multiple devices and user interfaces. Figure 1 shows the abstract PlugBot system architecture.

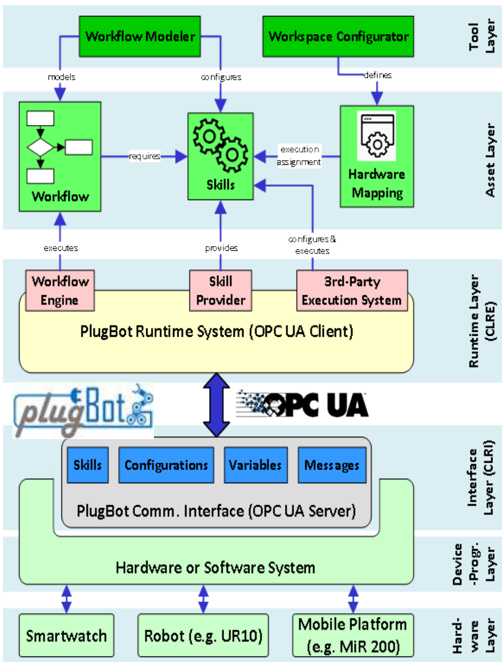


**Figure 1:** PlugBot System Architecture.

At the lowest level, the hardware layer, interfaces to hardware components are provided including proprietary software modules (e.g. on-board robot control and programming). The device programming layer encompasses the implementation of modular, reusable, type specific functionalities and realizes the abstraction based on system skills (see above). These software modules utilize the HW / SW systems of the hardware layer to apply the skill functionality in a process (step). In order to make these functionalities a) configurable, b) executable, and c) monitorable in a standardized way, a uniform interface for integration was developed in PlugBot and described in our previous work [14]. This interface follows the paradigm of skill-based programming. Hardware independence can be achieved through implementation of the standardized interface by a device from a new vendor.

The skill-based interface is implemented at the interface layer. This layer contains the communication via OPC UA, as well as software libraries, which facilitate and standardize the realization of skills and their integration as independent PlugBots. The interface layer deliberately uses OPC UA as a platform independent standard, fostering from its built-in features (discovery, secure communication, quality of service, semantic data models), to realize an integrated, modular system.

Workflows are executed on the runtime layer and represent sequentially chained skills. The execution of individual skills can also be triggered through third-party systems that act as OPC UA clients. Logical linking of a) a created workflow and b) the concrete PlugBot instances involved in the execution of this workflow (hardware mapping) is done within the asset layer. The basic idea is that a workflow can be created independently of the existing PlugBots. A logical assignment of concrete skill executions to PlugBots is done in a separate step. Finally, the tool layer covers development tools for workflows, workspace configuration (hardware mapping) and also application specific user interfaces (GUIs).

The next section describes an example use case of the developed system architecture for the application field "Mobile Manipulator". Note that application-specific architectures basically follow the generalized system architecture according to Figure 1, but do not necessarily include all its elements.

## 3.2. Application use case

Pick-up and delivery robots combine multiple systems with specific skills (transport, manipulation). Interoperability of robots with other machines enables automatic machine operation.

For the production of optical storage media, so-called stamper (dies) which contain the digital content of the customer's data are required and have to be transported between automatic storage systems and production machines. In order to maintain productivity a mobile manipulator consisting of an articulated robot (for object manipulation) as well as a mobile platform (to provide transportation skills within the plant) is intended to take over this task. Human operators are informed using smartwatch-based notification systems and take over complicated (de-)installation tasks of the stampers.

A second use case considers the assembly of customer-specific geared motors, where components have to be retrieved and moved to the assembly. To justify the equipment costs for mobile manipulators it is crucial utilizing the manipulator during transportation time for commissioning components (e.g. roller bearings) from delivery containers to dispensing units. A robot pick and place skill is deployed in parallel to the transportation skill of the mobile robot technically robust, safe and legally compliant.

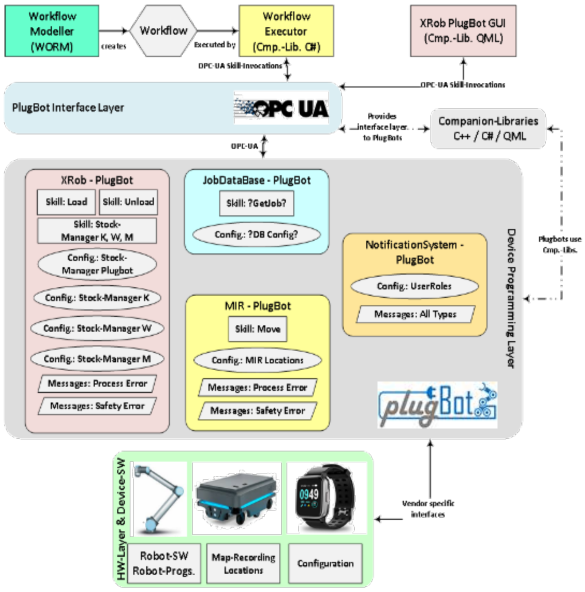


**Figure 2:** Mobile Manipulator System.

Figure 2 summarizes the load, unload and stock manager (part storage status tracking) skills for the articulated robot (implemented in and organized by XROB [28]), a movement skill for the MiR-type mobile robot, skills for operator notification as well as a JobDataBase skill to take over communication with a superior MES.

## 4. Conclusion

The PlugBot project focuses on the integration of robotic tasks in workflows that involve humans and multiple machines. Skills are used as a conceptual abstraction that support and facilitate the hardware agnostic integration and programming of systems. The developed architecture and use case implementations provide interoperability on technical level by using the standard OPC UA interface technology. Semantic interoperability is given by skills that serve as a unification concept on task level. Using a workflow engine supports the orchestration of the skills and therefore serves as interoperability tool on process and organizational level. The presented concept was implemented and evaluated in multiple use cases that showcase the practicability of the approach.

Two major findings were gained during the realization of the showcases by industry and research partners of the PlugBot consortium. Firstly, the developed skill-based (programming) interface helped to put focus on the implementation of skill behavior and workflows rather than dealing with integration aspects. Additionally, the skill implementation turned out to be less error prone due to standardized interfaces. Secondly, the PlugBot architecture is applicable to different application areas (e.g. mobile care systems, traffic management, grid management systems), targeting different heterogeneous systems that need to interact.

In an initial next step, the project ROBxTASK hast started, where focus is on a process modelling and workflow engine. It also contains skills as abstraction layer. End-users should be enabled to provide process descriptions that can then be executed by multiple human and artificial agents [29].

In more distant, future work, a superior system with automatic resource allocation and scheduling using a multi agent systems approach is planned [27]. The PlugBot concept will be applied to teams of mobile (robotic) platforms in order to reduce the workflow programming and debugging effort. To foster increased autonomy of the mobile robotic platforms (conceptualized as agents), further investigation of the PlugBot standardized interfaces is required to assess the applicability to realize typical multi agent system (MAS) functionalities like negotiation.

## 5. Acknowledgement

The research leading to these results has been developed in the "Plug and Produce Robotic Building Blocks" (PlugBot) project (Grant Nr. 874293) and the ROBxTASK project (Grant Nr.: 880866). Both projects have been funded by the Austrian Federal Ministry for Climate Action, Environment, Energy, Mobility, Innovation and Technology (BMK) and the Federal Ministry for Digital and Economic Affairs (BMDW) through the Austrian Research Promotion Agency (FFG).

## 6. References

[1] G. Weichhart, Y. Ducq, G. Doumeingts, An IFIP WG5.8 State-of-the-Art View on Methods and Approaches for Interoperable Enterprise Systems, in: M. Goediecke, E. Neuhold, K. Rannenberg (Eds.), Advancing Research in Information and Communication Technology: IFIP's Exciting First 60+ Years, Views from the Technical Committees and Working Groups, Springer, Cham, 2021, pp. 222–244. doi:10.1007/978-3-030-81701-5_9.

[2] F. Vernadat, Enterprise modelling: Research review and outlook, Computers in Industry 122 (2020) 103265. doi:10.1016/j.compind.2020.103265.

[3] R. Stamper, K. Liu, M. Hafkamp, Y. Ades, Understanding the roles of signs and norms in organizations - a semiotic approach to information systems design, Behaviour & Information Technology 19 (2000) 15–27. doi:10.1080/014492900118768.

[4] G. Weichhart, H. Panetto, A. Molina, Interoperability in the Cyber-Physical Manufacturing Enterprise, Annual Reviews in Control 51 (2021) 346–356. doi:10.1016/j.arcontrol.2021.03.006.

[5] L. von Bertalanffy, General System Theory - Foundations, Development, Applications, 17th ed., George Braziller, New York, NY, 1969.

[6] L. von Bertalanffy, An Outline of General System Theory, The British Journal for the Philosophy of Science 1 (1950) 134–165.

[7] Å. Fast-Berglund, D. Romero, M. Åkerman, B. Hodig, A. Pichler, Agent- and Skill-Based Process Interoperability for Socio-Technical Production Systems-of-Systems, in: Advances in Production Management Systems. Towards Smart and Digital Manufacturing. APMS 2020, Springer, Cham, 2020, pp. 46–54.

[8] A. Napoleone, M. Macchi, A. Pozzetti, A review on the characteristics of cyber-physical systems for the future smart factories, Journal of Manufacturing Systems 54 (2020) 305–335. doi:10.1016/j.jmsy.2020.01.007.

[9] A. Kusiak, Smart manufacturing, International Journal of Production Research 56 (2018) 508–517. doi:10.1080/00207543.2017.1351644.

[10] A. Kusiak, Fundamentals of smart manufacturing: A multi-thread perspective, Annual Reviews in Control 47 (2019) 214–220. doi:10.1016/j.arcontrol.2019.02.001.

[11] A. Köcher, C. Hildebrandt, B. Caesar, J. Bakakeu, J. Peschke, A. Scholz, A. Fay, Automating the development of machine skills and their semantic description, in: 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), IEEE, Vienna, 2020, pp. 1013–1018. doi:10.1109/ETFA46521.2020.9211933.

[12] S. Malakuti, J. Bock, M. Weser, P. Venet, … A. Bayha, Challenges in Skill-based Engineering of Industrial Automation Systems, in: 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), IEEE, Turin, 2018, pp. 67–74. doi:10.1109/ETFA.2018.8502635.

[13] F. Spitzer, R. Froschauer, T. Schichl, ATLAS-A Generic Framework for Generation of Skill-Based Control Logic and Simulation Models for Intralogistics Applications, in: 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), IEEE, 2021, pp. 1–8.

[14] F. Spitzer, R. Lindorfer, R. Froschauer, M. Hofmann, M. Ikeda, A generic Approach for the Industrial Application of Skill-based Engineering using OPC UA, in: 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), IEEE, Vienna, 2020, pp. 1671–1678. doi:10.1109/ETFA46521.2020.9211872.

[15] J. Pfrommer, D. Stogl, K. Aleksandrov, V. Schubert, B. Hein, Modelling and orchestration of service-based manufacturing systems via skills, in: Proceedings of the 2014 IEEE Emerging

Technology and Factory Automation (ETFA), IEEE, Barcelona, 2014, pp. 1–4. doi:10.1109/ETFA.2014.7005285.
[16] U. Thomas, G. Hirzinger, B. Rumpe, C. Schulze, A. Wortmann, A new skill based robot programming language using UML/P Statecharts, in: 2013 IEEE International Conference on Robotics and Automation, IEEE, Karlsruhe, 2013, pp. 461–466. doi:10.1109/icra.2013.6630615.
[17] OPC Foundation, OPC 40010-1 - Robotics Part 1: Vertical Integration, 2019. URL: https://opcfoundation.org/developer-tools/specifications-opc-ua-information-models/opc-unified-architecture-for-robotics/.
[18] M. Kaspar, J. Bock, Y. Kogan, P. Venet, M. Weser, U. E. Zimmermann, Tool and Technology Independent Function Interfaces by Using a Generic OPC UA Representation, in: 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), IEEE, Turin, 2018, pp. 1183-1186. doi:10.1109/etfa.2018.8502647.
[19] S. Profanter, K. Dorofeev, A. Zoitl, A. Knoll, OPC UA for plug produce: Automatic device discovery using LDS-ME, in: 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), IEEE, Limassol, 2017, pp. 1-8. doi: 10.1109/ETFA.2017.8247569.
[20] K. Dorofeev, C.-H. Cheng, M. Guedes, P. Ferreira, S. Profanter, A. Zoitl, Device adapter concept towards enabling plug&produce production environments, in: 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), IEEE, Limassol, 2017, pp. 1–8. doi: 10.1109/ETFA.2017.8247570.
[21] S. Profanter, A. Breitkreuz, M. Rickert, A. Knoll, A Hardware-Agnostic OPC UA Skill Model for Robot Manipulators and Tools, in: 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), IEEE, Zaragoza, 2019, pp. 1061-1068. doi: 10.1109/ETFA.2019.8869205
[22] G. Weichhart, D. Wachholder, On the Interoperability Contributions of S-BPM, in: A. Nanopoulos, W. Schmidt (Eds.), S-BPM ONE - Scientific Research, vol. 170, pp. 3–19, Springer International Publishing, Berlin, 2014, pp. 3-19. doi: 10.1007/978-3-319-06065-1_1.
[23] BKCASE Editorial Board, Guide to the Systems Engineering Body of Knowledge (SEBoK), 2021. URL: https://www.sebokwiki.org/w/images/sebokwiki-farm!w/9/90/Guide_to_the_Systems_Engineering_Body_of_Knowledge%2C_v._2.4.pdf
[24] F. B. Vernadat, Interoperable enterprise systems: Principles, concepts, and methods, Annual Reviews in Control, 31 (2007) 137–145. doi: 10.1016/j.arcontrol.2007.03.004.
[25] F. B. Vernadat, Technical, semantic and organizational issues of enterprise interoperability and networking, Annual Reviews in Control 34 (2010) 139–144. doi: 10.1016/j.arcontrol.2010.02.009.
[26] R. Froschauer, W. Kurschl, J. Wolfartsberger, S. Pimminger, R. Lindorfer, J. Blattner, A Human-Centered Assembly Workplace For Industry: Challenges and Lessons Learned, Procedia Computer Science 180 (2021) 290–300.
[27] G. Weichhart, Å. Fast-Berglund, D. Romero, A. Pichler, An Agent- and Role-based Planning Approach for Flexible Automation of Advanced Production Systems, in: 2018 International Conference on Intelligent Systems (IS), IEEE, Funchal, 2018, pp. 391–399. doi: 10.1109/IS.2018.8710546.
[28] S. C. Akkaladevi, M. Plasch, A. Pichler, Skill-based learning of an assembly process, E Elektrotechnik Informationstechnik, 134 (2017) 312–315, 2017. doi: 10.1007/s00502-017-0514-2.
[29] G. Weichhart, A. Pichler, F. Strohmeier, M. Schmoigl, H. Zörrer, The ROBxTASK architecture for interoperability of robotic systems, in: IEEE International Workshop on Metrology for Industry 4.0 & IoT (MetroInd4.0&IoT), IEEE, Rome, 2021, pp. 449-453. doi: 10.1109/MetroInd4.0IoT51437.2021.9488560.