Towards Easy Robot System Integration: Challenges and Future Directions

Daniella Tola[*,**], Emil Madsen[*], Cláudio Gomes[*], Lukas Esterle[*], Christian Schlette[§], Casper Hansen[†] and Peter Gorm Larsen[*]

***Abstract*—** **The amount of robots in industry is increasing, pushing the need for easier configuration and integration of robots. The cost of robot integration is often larger than the cost of the robots themselves. With the increasing complexity of robotic systems and the tasks they perform, the future of reducing the expenses of robot system integration is significant. In this position paper, we report on some of the main challenges in robot system configuration, and give an overview of the current state-of-the-art of robot system configuration. We briefly look into the configuration of personal computers (PCs) and study how the robot system configuration can be improved inspired by the standards used in the PC area. We conclude with concrete proposals as to what can be done in the future, to make the process of robot system configuration and integration easier and cheaper.**

I. INTRODUCTION

As the sales of smaller industrial robots and collaborative robots (cobots) is rapidly increasing together with the complexity[1], the need for easier methods for configuring such systems becomes important. Customers investing in robots are highly reliant on system integrators – businesses and individuals specialised in robot configuration and integration. System integrators help the customer choose valid configurations for particular applications.

A robot system, as defined in [1], consists of a robotic arm, an end-effector and other devices required to perform a desired task, as illustrated in **??**. Each of these devices are connected through matching interfaces and work in coordination to perform a particular task, such as screwdriving, machine tending etc.

Configurators are tools used to facilitate configuration tasks [2]. Robot systems and personal computers (PCs) are examples of systems that can benefit from configurators. They each compose different components that are validly connected to perform a particular task. Configurators have been developed for many years now, and are widely used in the PC domain, where component interfaces are standardised[2]. In contrast, configurators have not yet become common in robot systems. Instead, system integrators use documentation and vast experience to know which components are compatible. With the constant increase in deployment of robots in the industry, system integrators may face challenges in performing all demanded tasks.

Robotic kits are a way to do easy system integration, as all components in a kit are compatible. This presents a challenge for sustainability though, as kits *treat the symptom rather than the underlying cause* of configuration challenges.

Configurators for robot systems must be created in order to reduce the costs of deployment, lower the need of system integrators to the customers' advantage, and increase component reuse and therefore decrease waste. However, before this can be achieved, the challenges related to robot configuration must be overcome.

In this position paper we:

- Outline the related work in the area of robot system integration and configuration (Section II);
- Present detailed challenges related to robot system configuration (Section III);
- Summarise robot system and PC configuration and look at some of the main standards in this area (Section IV);
- Outline how we believe some of these challenges can be overcome, and how to move forward in the domain of robot system configuration (Section V).

II. RELATED WORK

Some of the challenges related to robot configuration and integration have already been described by Sanneman et. al. [3], which interviewed a number of leading robot manufacturers, research institutions working with robotics, and industrial manufacturing companies using robots. The main challenge related to robot configuration they presented is the missing standardisation of the hardware used in robot systems.

An outline for a user friendly platform for configuring robotic solutions is proposed in Schäffer et. al. [4]. They describe that creating configurators for robot systems is complicated due to a number of issues related to lack of standardisation and insufficient documentation. Their focus is on the proposition of a general robotics platform, and do not go into detail with specific robot systems and their devices.

In Schou et. al. [5] a prototype configurator was implemented for selecting suitable hardware for industrial cobots. They worked at an abstraction level related to the skills of the overall robot system, and did not look into configuration with regards to the compatibility of the devices within.

We would like to thank the Innovation Foundation Denmark for the MADE FAST project, and also thank the anonymous reviewers and Jonas Kjær Rask for the detailed feedback.

[*] Department of Electrical and Computer Engineering, Aarhus University, Aarhus, Denmark

[§] University of Southern Denmark, Odense, Denmark

[†] Technicon ApS, Hobro, Denmark

[**] Corresponding author: dt@ece.au.dk

[1]https://ifr.org/downloads/press2018/Presentation_WR_2020.pdf

[2]An example customisation tool: https://pcpartpicker.com

This paper complements the preliminary work carried out in [3], where we take a step further and present examples of specific challenges encountered during the configuration of various robot systems.

III. CHALLENGES

This section describes some of the main configuration and integration challenges encountered while assembling robotic cells in the Aarhus University Digital Transformation Lab[3], and during the development of a robot system configuration tool.

Each of the challenges below makes it difficult to configure robotic systems, and note that we frequently encountered more than one of these challenges in our integration tasks. The challenges and examples we describe originate from real industrial manufacturers, but to avoid harming their reputation we have chosen to anonymise their names.

A. Missing Standardisation of Interfaces

To successfully connect two devices together, we usually look at the interfaces and determine their compatibility. If the two interfaces fit together, they seem to be compatible. To determine the compatibility of the numerous devices in a robot system, we must know all their interfaces. Establishing which types of interfaces each device in a robot system support is not a simple task due to the difference in documentation of devices interfaces. Standardising these interfaces could make it easier when determining the compatibility of the devices. Currently, only the robot flange is standardised in *ISO 9409-1* [6], and the communication protocols used for different robots follow standards of Industrial Communication Protocols [7]. There are still standards between the device interfaces which are not yet created, for example a standard for describing the mechanical interface between the robotic arm and base. Each interface of each device should be specified, and if applicable a standard describing the types of the interfaces should be defined. One of the companies interviewed by Sanneman et. al. [3] stated that standardisation of both hardware and software is crucial for the robotics domain. Another company stated they were already working on standardising the hardware in their robot cells.

B. Standardised Flange Information not Available

The flange of a robotic arm is standardised in *ISO 9409-1* [6], and corresponds to the mechanical interface of the robotic arm where the EECD is attached.

Unfortunately, this standard is not always used in the specifications and data sheets of robotic arms: Table I shows numbers from examining specifications of robotic arms from three leading manufacturers. Only one of the robotic arm manufacturers has provided the ISO flange class of all their robotic arms, while another manufacturer has provided the ISO flange class of two out of eleven of their robotic arms. The third manufacturer has not provided the ISO flange class of any of their robotic arms.

[3]`https://projekter.au.dk/en/dtl/`

TABLE I: ISO flange information numbers.

# manufacturers	# robotic arms	# ISO flange class provided
3	34	8

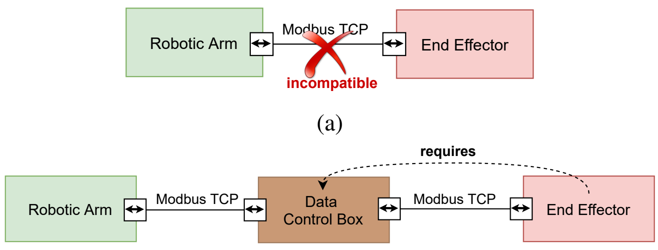

Fig. 1: Example of misleading compatibility specification, recreated from [8]. As illustrated in (a), both the Robotic Arm and End Effector use Modbus TCP as a communication interface, but are specified in the data sheet to be incompatible. Instead, as (b) shows, the End Effector requires a Data Control Box for connecting to the Robotic Arm.

The ISO flange class of the robotic arms could in principle be determined by examining the *ISO 9409-1* standard, which at the time of writing costs 38 CHF, and comparing the flange measurements described in the data sheet of the robotic arm, but this is a tiresome task.

C. Misleading Compatibility Specification

When two devices support the same industrial communication protocol [7], the communication between them is expected to be compatible. During the configuration of an end-effector and robotic arm, we found that the robotic arm supports Modbus TCP, and after contacting the manufacturer of the end-effector, we obtained a non-public document specifying that the end-effector supports Modbus TCP. As these two devices communicate with Modbus TCP, you would expect they are compatible. Unfortunately, this was not the case, as illustrated in Fig. 1. An extra requirement, specified in the obtained document from the manufacturer, described that the end-effector required to be connected with a data control box in order to be compatible with the robotic arm. It was also stated that the data control box supports Modbus TCP.

D. Missing Property Specifications

Different electrical end-effectors require different amounts of current to function. The electrical properties of such electrically actuated end-effectors should be described in a data sheet or technical report, for users to be able to determine which EECDs or robotic arms supply sufficient current. While looking into the different EECDs from one of the main manufacturers, we found that the maximum current that is supplied by the EECD was not documented in the data sheets, but instead one of the EECDs contained this value in its name. The example, illustrated in **??**, shows how the naming of the device was used to distinguish between two EECDs that provide a different amount of current. This is based on a real example, where an empirical test was performed with the *EECD* and end-effector and it was found that the two devices were incompatible, later discovering that the end-effector specifically required the *EECD 3A*.

E. Misleading Incompatibility Specification

Specifying that two devices are incompatible must mean that they should never be connected. An example where this is not true is illustrated in **??**. In the data sheet of an end-effector it was stated, in the section describing how to connect the device, that a specific robotic arm must never be connected to an EECD of a special type, which we will call *EECD B*. In another section of the end-effector data sheet, it was also specified that *EECD B* can be connected to the specific robotic arm, if a data control box was used instead of a data cable. This example shows that to fully understand the compatibility of a device the complete data sheet must be studied, which can be time consuming and in some cases confusing. Another problem with this is that if a device is described as incompatible, users that do not read through the complete data sheet may end up not reusing a device in a robot system due to lack of knowledge of compatibility.

F. Kits May Decrease Device Reuse and Sustainability

Various manufacturers of end-effectors provide *kits* that can be bought for specific series of robots. These kits contain the end-effector, the coupling device, the data connection, and any other necessary equipment for connecting the end-effector to a specific robotic arm. We found some cases where a kit was created for a specific type of robotic arm, but contained the exact same devices as another kit for another type of robotic arm, as illustrated in **??**. While these kits make it easier to purchase the right devices for a robot system, it does not promote reuse of devices, because buying a custom kit for a specific robotic arm is easier than determining which devices are needed in a new configuration and if they are compatible or not.

G. Summary

The consequences of the described challenges above is that companies are highly reliant on system integrators. When deploying robot systems, the main stakeholders and their connections are illustrated in **??**, which is similar to the deployment process described in [9]. The robot device manufacturers develop the different devices used in robot systems: base, robotic arm, EECD, end-effector, and data connection. System integrators use these devices in the configuration of robot systems for their customers, the robot system consumers. Here, the system integrator uses documentation and their prior experience in the configuration of robot systems. The robot system consumers use the robots in different areas such as manufacturing lines.

A number of companies that participated in the interviews performed by [3], describe that one of the main challenges they face is the integration of robots into a manufacturing line. The companies work with system integrators for accomplishing such tasks. In fact, the companies are so reliant on system integrators that some of them even informed they faced challenges with integrating the robots due to the scarce amount of system integrators. One of the companies in the interview claimed that their reliance on system integrators was a disadvantage, and therefore opted for in-house integration instead.

IV. Configuration of PCs and Robot Systems

In this section we describe the current development of configurators for PCs and robot systems, then we look into some relevant ISO standards developed for PC configuration for inspiration towards better configurators for robot systems.

A. Current Development in Configurators for PCs and Robot Systems

Configuring a PC nowadays is a fairly easy task, thanks to the existing PC configurators that can be accessed online, such as PC Part Picker[4]. These configurators would be difficult to implement if it was not for the developed standards within the domain of PCs. **??** shows a simple example of three devices within a PC: motherboard, CPU, and RAM. Each of these devices have mechanical and communication interfaces that must be compatible with the connected devices. The CPU connects to the motherboard through the standardised Socket AM4 interface, which describes both the mechanical and communication interface. The RAM connects to the motherboard through the standardised DDR4-1600 interface, which also describes both the mechanical and communication interface. An additional feature of RAM that can be included is error-correcting code (EEC) memory. This is developed in such a way, the RAM fulfils the DDR4 standard regardless if the EEC memory is supported or not.

Unfortunately, configuring a robot system is not as simple as a PC. To our knowledge, no mature public robot system configurators exist, leaving the task of configuration up to experienced system integrators. **??** illustrates how a robotic arm and end effector must be connected with compatible communication type and flange type. Currently there is no standard that defines both the communication and mechanical interface between a robotic arm and end-effector. The ISO 9409-1[6] standard describes the flange interface of robotic arms but, as described in challenge B above, it is not always used by robotic arm manufacturers. Industrial communication protocols [7], such as Modbus TCP and Ethernet/IP, are used for communicating with robotic arms but, as described in challenge C above, the compatibility of devices relying on these protocols is not always defined clearly, and in some cases an extra device is required.

The most similar industrial software, we have found, to a robot system configurator, are the cobot application builders developed by Universal Robots[5] and ABB[6]. These application builders help guiding the user to understand the different types of applications and installations the robotic arms can

[4]https://pcpartpicker.com/

[5]https://www.universal-robots.com/builder/

[6]https://applicationbuilder.robotics.abb.com

be used in. The Universal Robots application builder was introduced in 2018, and the ABB application builder was introduced in the beginning of 2021. While both of these application builders are great tools that help making robot integration easier for end users, they only focus on static configurations with manufacturer-specific robotic arms. They do not incorporate robotic arms from other manufacturers, and do not provide much information on the other devices (i.e. EECDs, data connections, etc.) required to complete the setup of the robot system.

B. Relevant ISO Standards for Configuration

By analysing a number of ISO standards related to PC configuration we found that ISO 13066-1 [10] describes requirements and recommendations on hardware interoperability, and also explicitly defines the responsibilities of device manufacturers. Some of the main responsibilities of device manufacturers included in the standard are:

- Use of standard interfaces,
- Support of standard functionality and commands,
- Provide information about non-standard functionality of devices.

Similar requirements and responsibilities could be posed to the device manufacturers in the robotics domain.

The end users of PCs vary in both age and background, showing the ease of use of these systems. Standards describing the user interfaces of PCs, such as ISO 9241-171 [11], are part of the reason for the high usability of PCs. The standard defines names and labels to be used in the user interface regardless of the operating system, and also describes standards for font size, placement, and display. To increase the use of robots, standards defining the vocabulary, as ISO 8373 [1], and defining standardised names and labels on user interfaces is critical. An example in the robotics domain where the vocabulary varies between manufacturers is the End Effector Coupling Device (from ISO 8373) with names such as "Quick Changer" and "Coupling".

V. FUTURE DIRECTIONS

In this section we describe some of the technologies that can be used to tackle some of the described challenges and some of the directions that can be worked towards in the future for the ease of robot system configuration and integration.

A. Default Reasoning for Handling Incomplete Information

Building up a knowledge base and a configurator for robot systems is a difficult task, due to some of the arbitrary compatibility rules between devices and the incomplete information about device interfaces. In some cases, the initial information about a product may be incorrect or incomplete, as shown in the examples presented above in Section III. Here, the initial assumption about a particular device may be changed, and the use of a default reasoning framework, such as Answer Set Programming (ASP) is advantageous [12]. Default reasoning is described by [13] as an *exception-permitting generalisation*, it allows both to describe general and specific rules about the compatibility of devices. The different robot devices can be modelled by describing their interfaces (mechanical, data, electrical) and defining rules for creating configurations of devices with compatible interfaces. If a part of the interface information of a device is missing, as shown in example III-D, additional rules on, for example, the incompatibility can be added to the framework. With all the complicated rules within the robot system configuration domain, using default reasoning may be a benefit when developing a configurator. A prototype implementation of an ASP-based robot system configurator is presented in [8], showing this may be a fruitful direction.

B. Co-simulation for Virtual Scenario Assembly

Working towards usability in the robot configuration and integration domain, application builders such as the ones described in Section IV-A, could help end users better understand how robots can be integrated in their manufacturing lines and also which types of applications or robots that can be used. Virtual assembly of these devices can be difficult, especially if models of each device must be created separately. Instead, utilising co-simulation together with the Functional Mock-up Interface (FMI) [14] standard could be an opportunity. The FMI standard is one of the most common standards used for co-simulation, and allows encapsulation of models as black boxes. Different original equipment manufacturers (OEMs) can provide black box models of their devices using the FMI standard, and at the same time protect their Intellectual Property. These black box models can then be used in a co-simulation, where virtual scenarios of robot systems can be assembled. Co-simulation allows simulating these FMI-based black box models which can be developed using different tools [15]. Apart from virtual scenario assembly, it may be a possibility to use co-simulation to detect incompatibilities between the models of devices, as the FMI standard also focuses on interfacing inputs and outputs which already need to be matched up.

C. Explicit Responsibilities of Device Manufacturers

Currently, device manufacturers have the ability to create arbitrary rules for the connectivity of their products, and the system integrators then adapt their knowledge to these. But, to make it easier to understand the compatibility of devices within a robot system, it is necessary that the device manufacturers can also be held responsible, in order to boost ease of configuration. A future objective may be to define clear and explicit requirements and responsibilities for the device manufacturers of robot systems, similar to the ISO 13066-1 [10] in the PC configuration domain.

D. Creating Simple Device Standards

Looking at the standards defined in the PC domain, they incorporate both the communication and mechanical interfaces between devices. While this may seem inflexible in some cases, it allows for easier configuration, as the example presented in Section III-E is dependent on both the communication and mechanical interface. Creating standards that combine both the communication and mechanical interfaces of robot devices can potentially ease the process of configuration[7].

To easily configure systems, clear, standardised interfaces must be defined, but this also reduces the ability to define specialised applications. There is a trade-off between the flexibility of the robot system application and ease of configuration. Instead, with the vast experience that system integrators have, they can use their expertise to deal with specialised applications.

E. Application Packages

Easy configuration of robot systems can be facilitated by the use of *application packages*, which allow end-users to purchase a set of robot system devices to solve a given automation task. The compatibility of robot system devices in an application package is validated by one or more of the robot device manufacturers. This allows end-users to purchase a set of devices without needing to consult a system integrator to address the risk of device incompatibility. Examples on application package are those acquired through the Universal Robots UR+ platform[8] or the KUKA ready2_use[9] packages.

Examples on application packages include among the robot system devices;

- bases to dynamically displace the robot, e.g. linear motion kits and mobile robots,
- EECDs, e.g. the flanges for 1) quick and tool-less change of end-effectors and 2) adding force/torque measuring capabilities,
- software solutions, such as AI-based robotic arm health monitoring, welding and palletizing software, and
- other peripheral devices, e.g. conveyor tracks and safety guards such as light curtains.

These application packages are, however, constrained to work on robotics arms by a single manufacturer only. Such application packages certainly make it easier and less risky for end-users to acquire a set of robot system devices to solve a given automation task, but they are merely a step towards the goal of having general purpose robot system configurators with no constraint on the manufacturer of the robotics arms.

VI. CONCLUSION

Configuring and composing a robot utilising a variety of components is already a challenging task. With the rising number of robot component manufacturers and concomitant products, this task becomes more complex and with interfaces, allowing for components from different manufacturers being used together, the potential combinations increase even further. While standards exist, to harmonise interfaces, these standards are barely used by the manufacturers. This calls for integration approaches in the robot domain that push robot configuration towards being as easy as configuring a PC. In this paper, we outlined some of the main challenges faced when configuring and integrating robot systems. We drew a comparison between PC configuration and configuring robots and highlight approaches available by individual manufacturers. However, as current efforts are driven by and tailored to the products of individual manufacturers, there is need for a push towards a holistic approach supporting system integrators to easily combine elements produced by different producers. To achieve this, we propose future research directions including the usage of default reasoning frameworks such as Answer Set Programming and adapted processes from the PC domain to improve configuration of robots. We also propose co-simulation in virtual settings, not only enabling system integrators to experience robotic setups in their personalised deployment environments but also to identify compatibility and detect potential incompatibilities among the individual devices. This can be achieved without having the physical robotic devices on-site.

[7] As one of the reviewers pointed out, this is more challenging when dealing with interfaces that bear loads or can be subject to impacts.

[8] `https://www.universal-robots.com/plus/`

[9] `https://www.kuka.com/en-us/products/robotics-systems/kuka-ready2_use`

REFERENCES

[1] "Manipulating industrial robots – vocabulary," International Organization for Standardization, Geneva, CH, Standard, 2013.

[2] D. Sabin and R. Weigel, "Product configuration frameworks-a survey," *IEEE Intelligent Systems and their Applications*, vol. 13, no. 4, pp. 42–49, 1998.

[3] L. Sanneman, C. Fourie, and J. A. Shah, "The state of industrial robotics: Emerging technologies, challenges, and key research directions," *CoRR*, vol. abs/2010.14537, 2020. [Online]. Available: https://arxiv.org/abs/2010.14537

[4] E. Schäffer, M. Bartelt, T. Pownuk, J.-P. Schulz, B. Kuhlenkötter, and J. Franke, "Configurators as the basis for the transfer of knowledge and standardized communication in the context of robotics," vol. 72, pp. 310–315, 2018, proceedings of the 51st Conference on Manufacturing Systems.

[5] C. Schou, M. Hansson, and O. Madsen, "Assisted hardware selection for industrial collaborative robots," *Procedia Manufacturing*, vol. 11, pp. 174–184, 2017, proceedings of the 27th International Conference on Flexible Automation and Intelligent Manufacturing.

[6] "Manipulating industrial robots – mechanical interfaces – part 1: Plates," International Organization for Standardization, Geneva, CH, Standard, 2004.

[7] "Industrial robot communication protocols," https://s3.amazonaws.com/RobotiqContent/Documents/Industrial-robot-communication-protocols.pdf, (Accessed: 05-06-2021).

[8] D. Tola, C. Gonçalves Gomes, C. Schultz, C. Schlette, C. Hansen, and L. Esterle, "Robocim: Towards a domain model for industrial robot system configurators," in *Proceedings of the 15th International Rule Challenge, 7th Industry Track, and 5th Doctoral Consortium*, ser. CEUR Workshop Proceedings, A. Soylu, A. Nezhad, N. Nikolov, I. Toma, A. Fensel, and J. Vennekens, Eds. CEUR-WS.org, 2021.

[9] C. Schlegel, A. Lotz, M. Lutz, D. Stampfer, J. F. Inglés-Romero, and C. Vicente-Chicote, "Model-driven software systems engineering in robotics: Covering the complete life-cycle of a robot," *it - Information Technology*, vol. 57, pp. 85 – 98, 2015.

[10] "Information technology — interoperability with assistive technology (at) — part 1: Requirements and recommendations for interoperability," International Organization for Standardization, Geneva, CH, Standard, 2011.

[11] "Ergonomics of human-system interaction – part 171: Guidance on software accessibility," International Organization for Standardization, Geneva, CH, Standard, 2008.

[12] V. Lifschitz, "What is answer set programming?" in *Proceedings of the 23rd National Conference on Artificial Intelligence*, ser. AAAI'08. AAAI Press, 2008, p. 1594–1597.

[13] R. Koons, "Defeasible Reasoning," in *The Stanford Encyclopedia of Philosophy*, E. N. Zalta, Ed. Metaphysics Research Lab, Stanford University, 2017.

[14] Modelica Association, "Functional Mock-up Interface for Model Exchange and Co-Simulation," https://www.fmi-standard.org/downloads, October 2019.

[15] C. Gomes, C. Thule, D. Broman, P. G. Larsen, and H. Vangheluwe, "Co-simulation: A survey," *ACM Comput. Surv.*, vol. 51, no. 3, May 2018. [Online]. Available: https://doi.org/10.1145/3179993