CPS PLATFROM APPROACH TO INDUSTRIAL ROBOTS: STATE OF THE PRACTICE, POTENTIALS, FUTURE RESEARCH DIRECTIONS

Martin Mikusz, Graduate School of Excellence advanced Manufacturing Engineering, GSaME, University of Stuttgart, Germany, martin.mikusz@gsame.uni-stuttgart.de

Akos Csiszar, Graduate School of Excellence advanced Manufacturing Engineering, GSaME, University of Stuttgart, Germany, akos.csiszar@gsame.uni-stuttgart.de

Abstract

Approaches, such as Cloud Robotics, Robot-as-a-Service, merged Internet of Things and robotics, and Cyber-Physical Systems (CPS) in production, show that the industrial robotics domain experiences a paradigm shift that increasingly links robots in real-life factories with virtual reality. However, despite the growing body of research to date, though insightful, the paradigm shift to CPS in industrial robotics remains an under-researched area. Findings from the present paper make several contributions to the current state of research: We provide a potentially reusable framework of analysis and apply this framework in order to reveal whether and to what extent the industrial robotics branch implements abilities and characteristics of CPS. We examine the top five industrial robot manufacturers ABB, Fanuc, Kawasaki, Kuka, and Yaskawa and identify considerable, current implementations. However, concerning one of three perspectives—the perspective on CPS as industry platform constructs, takes the industrial robotics branch only certain small steps towards CPS platforms. We discuss them and outline a set of business model patterns that can transform product innovations, enabled by abilities and characteristics of CPS, into business model innovations in the industrial robot domain. In order to enable the industry to exploit the full potential of industrial robots understood as CPS, we question the right degree of openness in the context of industry platform constructs. Our methodological approach combines conceptual with empirical research.

Keywords: industrial robot, cyber-physical system, industry platform

1 INTRODUCTION

Nowadays, we can observe a shift of value creation towards diverse, cross-domain cooperating business networks. Customer-oriented business models, characterized by interactive value creation with customers and other external actors as well as innovation processes that are realized in inter-organizational networks are becoming key competitive factors. This facilitates cross-industry innovations, which ignore existing market boundaries and accelerate the convergence of previously separate markets. Businesses including mechanical engineering companies of varying sizes and industry segments increasingly cooperate with each other and with service providers, telecommunication suppliers, and software producers, in order to merge their competences that they eventually need to construct and operate cross-industry product innovation (Broy et al. 2012, Kagermann et al. 2014, Geisberger and Broy 2015).

As a consequence, new forms of cooperation and competition as well as new shapes of solutions with a declining share of mechanics and hardware on the overall customer value proposition are emerging. We understand cyber-physical systems (CPS), i.e. integrations of computation and physical processes, where physical processes affect computations and vice versa (Lee 2006, Lee 2008), as such a solution. Particularly, previously isolated products as well as business models of the traditional goods-producing industry melt together with those of software businesses. Broy et al. (2012) characterize CPS according to their prospective evolution towards increased openness, complexity, autonomy and intelligence or "smartness" by five capability levels built on one another:

- CPS fuse the physical and virtual world, especially by using intelligent sensors and actuators as well as real-time control systems;
- Depending on task and situation, CPS (temporarily) build systems of systems with dynamically changing system boundaries;
- CPS can adapt to the environment and the according requirements (context-adaptability) and thereby act fully or partially autonomous;
- CPS are globally interconnected, cooperative systems with distributed and possibly changing control;
- Extensive human-system cooperation becomes an inherent part of CPS—they are systems, in which humans are passive or active elements of the system behaviour.

Although industrial robots are inherently predestined to be understood as CPS, research and practice mostly refer to other industries, especially to the automotive sector, when they illustrate the potential of CPS (Broy et al. 2012, Kagermann et al. 2014, Geisberger and Broy 2015). Industrial robots and robotic systems are key components of automation within the application areas of handling operations, welding and soldering, dispensing, processing, assembling and disassembling etc. (IFR Statistical Department 2013). ISO 8373 (ISO 2012) defines an industrial robot as

- automatically controlled,
- reprogrammable (programmed motions or auxiliary functions may be changed without physical alterations),
- multipurpose (capable of being adapted to a different application with physical alterations, i.e. alteration of the mechanical structure or control system except for changes of software)
- manipulator programmable in three or more axes (direction used to specify the robot motion),
- which can be either fixed in place or mobile for use in industrial automation applications.

According to the World Robotics 2013 Industrial Robots study (IFR Statistical Department 2013), one of the major technical robotic trends and challenges at the same time is the Internet of Things (IoT)—a paradigm shift that increasingly links the real-life factory with virtual reality. In this paradigm, each automation device seamlessly communicates in a system of decentralized control of production with hitherto unknown levels of flexibility (IFR Statistical Department 2013). IoT is accompanied by the initiative Industry 4.0 that—beyond the use of IoT—also involves integration of CPS and cloud-based services into manufacturing and logistics (Kagermann et al. 2013). There is a close connection between Industry 4.0, IoT and CPS; one can put these approaches on the same level.

Latest research activities in the field of (industrial) robotics give attention to CPS-related aspects as well. The concept of Robot as a Service (RaaS) enforces the design and implementation of a robot to be an all-in-one SOA (Service-Oriented Architecture) unit. The idea behind this is that the RaaS should have complete functions of SOA—i.e. each unit hosts a repository of services. It can be used by this robot as well as shared with other robots. RaaS units provide standard interfaces to the environment. Thus, a developer or client can compose new applications based on the services available in- and outside the unit (RaaS cloud) (Chen et al. 2010, Chen and Hu 2013). Kehoe et al. (2015) provide a survey covering over 150 references on results and open challenges on Cloud Robotics. The survey considers robots that rely on data or code from an internet-accessible cloud infrastructure to support their operation, i.e., where not all sensing, computation, and memory is integrated into a standalone system. This survey is organized around four potential benefits of the Cloud: Big Data (e.g. access to libraries of trajectories), Cloud Computing (e.g. access to simulation tools on demand), Collective Robot Learning (e.g. robots sharing control policies), and Human Computation (e.g. use of crowdsourcing to tap human skills for error recovery) (Kehoe et al. 2015).

CPS-related aspects have been recently recognized as an important issue in research on industrial robotics. However, despite the growing body of research to date, though insightful, the paradigm shift to CPS in industrial robotics remains an under-researched area. The purpose of this paper is threefold:

- First, to reveal whether and to what extent the industrial robotics branch implements abilities and characteristics of CPS. Our framework of analysis shows it from three distinct perspectives;
- Second, to reveal untapped potentials. We assume that the full potential of industrial robots understood as CPS according to our framework of analysis has been not exploited yet;
- Third, to provide future research directions. We assume that research will be required in order to enable industry to exploit the full potential of industrial robots understood as CPS.

Our methodological approach combines conceptual with empirical research. Our conceptually derived framework of analysis (section 2) consequently draws on the abilities and characteristics of CPS and their underlying technical platforms. To examine the state of practice in the industrial robot domain, we systematically apply our framework (section 3). We gain data by structured analysis of publicly available documents (online available product information, press releases, annual reports etc.) of the analysed top five industrial robot manufacturers. The statistical database Statista offers an overview of the leading companies in the global industrial robot market in 2011. Yaskawa producing the Motoman robots has the largest market share with 23%, followed by Fanuc with 22%, Kuka with 15%, ABB with 13% and Kawasaki with 8%. Thus, our analysis covers a combined market share of 81% (Statista 2015). In section 4, we discuss the current state of the practice—grounded in our analysis in section 3—as well as untapped potentials and future research directions.

Findings from the present paper make several contributions to the current state of research:

- Industrial robots potentially possess all necessary abilities and characteristics of CPS according to our framework of analysis. As to both the perspectives on CPS as physical goods improved by properties of software and CPS as opened and linked-up systems in contrast to embedded systems, we identify considerable, current implementations in the industrial robot domain. However, concerning the perspective on CPS as industry platform constructs, takes the industrial robotics branch only certain small steps towards CPS platforms;
- Understanding industrial robots as CPS in the sense of industry platform constructs enables innovative, complementary products or services, and the use of industrial robots in innovative ways. It also opens up considerable potentials for innovative business models. We outline a set of business model patterns that can transform product innovations, enabled by abilities and characteristics of CPS, into business model innovations in the industrial robot domain;
- In order to enable industry to exploit the full potential of industrial robots understood as CPS, we question the right degree of openness in the context of industry platform constructs.

2 FRAMEWORK OF ANALYSIS

According to the current state of research, the present predominant understanding of CPS is technical and not driven by business- or service-related abstractions from the economic perspective (Mikusz 2015). Very briefly, CPS are considered to be a confluence of embedded systems, real-time systems, and a variety of distributed smart sensor systems and actuators (Rajkumar et al. 2010). This understanding provides the right abstractions to carry out solutions for technical or engineering challenges, but less for the purpose of this paper. In contrast to the current state of research, our framework of analysis goes beyond pure technical abstractions. It draws on abilities and characteristics of CPS from three perspectives (with slight modifications also applied in Mikusz et al. (2015)):

- CPS as physical goods improved by properties of software;
- CPS as opened and linked-up systems in contrast to embedded systems;
- CPS as industry platform constructs.

2.1 CPS as physical goods improved by properties of software

CPS are a new form of solutions with a reduced share of mechanics and hardware and instead an increased share of software and software-enabled services on the overall customer value proposition. Thus, CPS or previously pure physical goods at least partially exhibit characteristics of software from an economic perspective. A considerable part of CPS' overall customer value proposition can be directly attributed to the CPS' software part.

Research in the field of software business has pinpointed the specific economic properties of software for a long time. There is broad consensus about the dissimilarity between software and its value chain on the one hand, and services or physical goods on the other hand (Buxmann et al. 2013). E.g., Pussep et al. (2012) map the economic principles of software industry to attributes of software value chain activities. Schief and Buxmann (2012) develop a software industry business model framework comprising 20 elements, based on 43 specific economic properties of software that they retrieve from other sources. The software economic property "ease of replication", for instance, refers to 14 business model elements—e.g. the pricing model, the degree of standardization and the operating model are highly dependent from this economic property. The following selection of these properties is fundamental to our framework of analysis:

- Ease of replication, modification and individualization;
- High opportunities for differentiation;
- High importance of intermediaries, platform concepts, and standardization (sections 2.3 and 3.3).

2.2 CPS as opened and linked-up systems in contrast to embedded systems

CPS enable a wide range of novel functions, services, and features that are far beyond the scope of today's capabilities of externally non-networked embedded systems with controlled behaviour. An embedded system is a combination of hardware and software that is built into a larger mechanical or electrical system—ranging from toys to medical equipment. Embedded systems perform a narrow range of pre-defined, dedicated functions with very specific requirements. They are either invisible to the user, especially in the case of autonomous systems, or just assist him with the automated performance of tasks, but they cannot be influenced by the user. In latter case, embedded systems usually allow only controlled intervention through precisely defined interaction interfaces (assistance systems). In contrast to this, CPS are opened and linked-up systems that merge the physical and the virtual world through intelligent sensors and actuators. CPS gather information about environmental and system conditions in real time and react to this information interactively and cooperatively (Broy et al. 2012, Geisberger and Broy 2015). All in all, an industrial robot, which in turn is interpreted as a CPS, includes connected sub systems that

- immediately collect physical data by means of sensors (proximity sensors, angle and rotation sensors, electric current sensors, temperature sensors, further external sensors etc.),

- combine those data with additionally available data and services (remote diagnosis and maintenance, vision systems etc.),
- and interact on this basis actively or reactively with the physical and the digital world, including interaction with other CPS—this interaction takes place by means of actuators acting on physical processes via system interfaces as well as via human-machine interfaces (e.g. motion coordination of robots, human robot interaction etc.).

2.3 CPS as industry platform constructs

Both, from technical and business perspective, CPS require platform constructs or conceptualizations. Here, we refer to Gawer's (2014) classification of technological platforms and adopt the industry platform conceptualization as the underlying technical platform for CPS. Gawer (2014) defines industry platforms as products, services or technologies developed by one or more firms that serve as a foundation, upon which a larger number of firms organized as a business ecosystem can build further complementary products, technologies, or services. With slight modifications to Gawer (2014), we draw on abilities and characteristics of industry platforms from four perspectives:

- platform leader and complementors as constitutive agents;
- network-affected ecosystem governance;
- open interfaces and potentially unlimited pool of accessible innovative capabilities;
- modular design with core and periphery in order to improve efficiency and reduce cost.

Besides complementors, the second constitutive agent of an industry platform is the platform leader or owner of a platform, who drives industrywide innovation for an evolving system of separately developed complementary components (Gawer 2014).

Industry platforms operate within the broad organizational setting of the ecosystem, whereby coordination is ensured by ecosystem governance. In this regard, platforms are distinct in that they are associated with network effects. This means that there are increasing incentives for more developers of complementary products and users to adopt a platform and join the ecosystem as more users and complementors participate. Network effects can be very powerful and direct or same-side between the platform and the user of the complementary innovation, or between platform and the complementors (Gawer 2014, Gawer and Cusumano 2014).

Industry platforms have open technological interfaces, but there are variations within the spectrum of how open these interfaces are. Potential innovators of complementary products can utilize information on the platform's technological interfaces that are disclosed by the platform leader to build compatible complements. Industry platforms therefore widen the pool of accessible innovating agents and their innovative capabilities to a potentially unlimited extent (Gawer 2014, Gawer and Cusumano 2014).

Across all organizational settings or levels—firm, supply chain or business ecosystem—all kinds of platforms have a modular architecture organized around a core and a periphery, from which a stream of derivative or complementary products, technologies, or services can efficiently be developed and produced (Gawer 2014). Thus, (industry) platforms provide a foundation of reusable common components or technologies. The objective is to improve efficiency and reduce cost by systematic reuse of modular components (Gawer and Cusumano 2014).

3 ANALYSIS OF THE INDUSTRIAL ROBOT DOMAIN

3.1 Industrial robots as CPS in the sense of physical goods improved by properties of software

In case of a robot, software is an essential part of the product. The control system is responsible for assuring the desired behaviour of the robot; the robot program, a script-like software component, defines this desired behaviour. The robot program is written either by the end-customer or by robot integrator companies, which are responsible for delivering the robot integrated in a robot cell (or robotic system) to the end-customer. All of the top five robot manufacturers provide the necessary software tools—development and simulation environments—to carry out programming of the robot, i.e. to customize it for the given task it is supposed to carry out. Only ABB goes as far as offering an SDK (software development kit) that can be used for developing add-ons for its simulation and programming tool RobotStudio, as well as for its control software RobotWare. In other words, ABB permits also customization of its programming environment. Furthermore, from the mentioned top five, only ABB facilitates the implementation of add-ons like components from third parties. This possibility is only recently available (since Q4 2014).

Four of the top five manufacturers offer software packages for the application of domain-specific tasks (e.g. handling, welding) or for specific features (e.g. multi robot coordination). These software packages expand the robot program functions and its initial or general instruction set. Software packages are similar to add-on libraries and contain an additional set of instructions that can be used during the programming of the robot for a specific task. Only Yaskawa does not mention software packages or optional software for their industrial robots. Kawasaki offers all of its optional software packages for free, except one, the collision detection software package. The other three offer software packages for purchase. A practical example for such packages is the software package for handling tasks (Fanuc's "Coordinated motion package", "KUKA.ConveyorTech", ABB's "Motion Coordination"). All of these software packages include instructions to synchronize the movement of the robot with the movement of a conveyor belt. Finally, four out of the top five manufacturers, with exception of Kawasaki, offer the possibility to implement customized screens on the human machine interfaces—also called control pendants—of their robots (Kuka Conveyortech, Fanuc Coordinated Motion Package, ABB SDK/RobotStudio Add-Ons, ABB Robot Apps, Yaskawa Advanced PP Customization SDK, ABB Robot Application Builder, Kuka Control & Observe, Kuka.HMI, Fanuc Controller R-30iB).

All of the top five industrial robot manufacturers use the probably most powerful software-related differentiation opportunity: Currently, all of them offer their own proprietary programming language (domain specific languages). Recently, new approaches using General Purpose languages (GPL) can be observed. Yaskawa is offering the possibility to program its new robot controller in C or C# using its SDK called Motoman SDK. Kuka plans to offer the possibility to program its upcoming Sunrise Controller in JAVA using the Sunrise API (Kuka 2014, Yaskawa FS100 Controller). By using different SDKs the manufacturers make sure that even in the case of a command GPL type language there will be no cross-compatible source code and, therefore, the "lock-in" effect of existent knowledge and investments in products of one manufacturer will continue to exist. An important step towards a modern software architecture that offers flexible integration possibilities on larger software systems is the SOA approach recently introduced by ABB. Since the interface is strongly manufacturer-specific, even if other manufacturers decide on similar approaches, we expect the differentiation aspect to be preserved.

We observed also another trend worth noting in respect of differentiation and programming of robot controllers: Instead of requiring that their industrial robots were to be programmed in their own programming environments, robot manufacturers allow that other industrial automation equipment manufacturers use their own proprietary automation controller software also to program robot controllers. An example for this is the option to program some of the robot controllers offered by Yaskawa using the programming environment of the automation equipment manufacturer Rockwell Automation. By using the optional software package Kuka.PLC, the Kuka robot does not require a robot program anymore; instead it accepts motion commands from different automation controllers. This option is available for automation equipment manufacturers Siemens and Rockwell, as well as for other automation equipment manufacturers using the CODESYS software development platform. These automation controllers are programmed in their proprietary programming environment. Kuka offers another feature, similar in nature: Kuka robots can be also controlled by CNC controllers, not just general-purpose automation controllers. Robots often have the task of loading and unloading machine tools controlled by such CNC controllers. In order to assure a seamless integration in this task, Kuka robots can be controlled by Siemens CNC controllers. The other three robot manufacturers do not offer such features (Kuka.PLC mxA, Yaskawa Program Motoman Robots with Rockwell Automation PACs).

Finally, we observed a certain degree of differentiation among the aforementioned software packages. Most of the top five manufacturers offer the same basic features in software packages, just with different names and implemented only with minor differences. However, Kuka is currently the only manufacturer who offers the integration of CNC Machine tool specific G-Code (machine tool source code) through its software package KUKA.CNC. ABB is also offering a machining software package, but there is no option for integrating G-Code. Yaskawa has recently collaborated with other companies to develop a solution for robot machining but no further details are known. The other two manufacturers offer no software tools to assist in developing robot-machining applications. As for process monitoring, ABB distinguishes itself from the other top five manufacturers by offering a software tool to visualize and log process data with its software tool Webware. Furthermore, ABB built its new control software (released Q4 2014) as a service-oriented architecture (SOA), with available RESTful web services (Kuka.CNC, ABB RobotStudio Machining PowerPac, ABB Robot Web Services, ABB WebWare Server).

Table 1 outlines the results of our analysis from the perspective of this chapter.

CPS' abilities and characteristics in general	**Current implementation in the industrial robot domain**	**ABB**	**Kuka**	**Kawa-saki**	**Yas-kawa**	**Fanuc**
Software-enabled configuration, modification, and individualization capabilities	Robot program & software tools for robot programming	✓	✓	✓	✓	✓
	SDKs for developing of add-ons	✓	-	-	-	-
	Implementation of add-on like components from 3rd parties	✓	-	-	-	-
	Addidional software packages for application of domain specific tasks	✓	✓	✓	-	✓
	Customized screens on human machine interfaces	✓	✓	-	✓	✓
Software-enabled high opportunities for differentiation	Own proprietary programming language	✓	✓	✓	✓	✓
	GPL with proprietary SDKs	✓	-	-	✓	-
	Control using 3rd party industrial controller	-	✓	-	✓	-
	Niche applications & software packages	✓ / -	✓	-	-	-

Table 1. Industrial robots as CPS in the sense of physical goods improved by properties of software

3.2 Industrial robots as CPS in the sense of opened and linked-up systems in contrast to embedded systems

Industrial robots have several internal built-in sensors that are required for their correct functioning—among others proximity sensors, angle and rotation sensors, electric current sensors and temperature sensors collect physical data. All top five manufacturers support interfacing of further external sensors through communication protocols. The protocols for interfacing the periphery are open or partially open. A main external information source is sensor data coming from vision systems. Integrating vision systems is particularly important to enable flexibility with regard to uncertainties in the environment of the robot. With integrated vision systems, robots are able to react to scenarios that cannot be predefined or would be too expensive to predefine. A practical example for this is welding of work pieces that have high positioning tolerances. Using vision systems, the starting point of a welding seam must not be hard-coded (requiring submillimeter work-piece to robot root positioning tolerances), but can be identified by the vision system (eliminating work-piece to robot root positioning tolerances). All top five manufacturers offer integrated vision systems.

Furthermore, all of the top five manufacturers offer the OPC UA communication protocol in order to enable interoperable communication or data exchange, e.g. with a manufacturing execution system as external data source. I.e., they offer the possibility to access parameters of the robot controller. OPC UA is a well-established communication protocol in process control for non-real-time critical data access. This protocol can be used to cyclically monitor or log process variables in the robot controller. Furthermore, ABB offers WebWare, a software tool that allows the user accessing, logging, and visualizing the robot's behaviour or mission critical data coming from the robot over a communication network. In a sense, WebWare implements telemetry access to ABB's industrial robots. No other manufacturer is offering a similar software tool, although they allow access to similar data through the OPC UA communication protocol. However, in these cases, the telemetry software tool has to be custom-made by the user himself or a third party solution has to be customized. Remote diagnosis by the manufacturer himself is available from all of the top five manufacturers. Remote access via a network connection to the robot reduces maintenance costs and eases technical support (Yaskawa. Total Customer Support Overview: Service, Maintenance, Spare Parts, Repair, Training, Fanuc Remote Diagnostics, Kuka Remote Control, Kawasaki E-Controller, ABB Maintenance).

Interacting with the real world is an inherent part of a robot's task—all tasks of industrial robots have as a goal to make some kind of change in their environment. This change can mean to manipulate an object, to weld work-pieces together or similar. In case of cooperating robots, two or more robots interact with their physical environment and with each other while carrying out their tasks. This direct interaction enables features that would not be possible without a direct digital communication. A practical example for this is motion coordination of robots where cooperating robots exchange pose information in real-time in order to enable synchronized relative motions and collaborative performance of tasks. All of the top five manufacturers offer collaboration features for their robots. Collisions are undesired interactions with the physical world, including other robots. Detecting collisions with surroundings is an essential feature in modern robotic applications heavily reliant on (in most cases internal) sensors. Collision detection is a reactive measure minimizing after-impact damages. All of the top five manufacturers offer collision detection packages.

For human robot cooperation or physical interaction, not detection but collision avoidance strategies are required—again features that are heavily reliant on sensors (internal and/or external safety-related sensors). If a human is present in the (pre-) defined safe zone inside the robot cell, the robot has to maintain a safe velocity or come to a controlled full stop. The decision if a full stop has to be carried out is based on the distance of the robot to the safe zone—evaluated by internal safety-related sensors. The governing robotic safety standard ISO 10218 (ISO 2011) permits physical interaction between a human and a robot—e.g. a person comes in direct contact with the robot for loading/unloading a work-piece to/from the gripper or tool of the robot—only in the case when the robot is in a safely stopped state. All top five manufacturers offer safe human robot cooperation packages complying with governing standards (ISO 10218 or equivalent).

Table 2 outlines the results of our analysis from the perspective of this chapter.

CPS' abilities and characteristics in general	**Current implementation in the industrial robot domain**	**ABB**	**Kuka**	**Kawa-saki**	**Yas-kawa**	**Fanuc**
Immediate collection of physical data by means of sensors	Internal sensors: proximity, angle and rotation, temperature sensors; further external sensors	✓	✓	✓	✓	✓
Combination of collected sensor data with additional available data and services	Vision systems (external)	✓	✓	✓	✓	✓
	Interoperable communication or data exchange via OPC UA protocol	✓	✓	✓	✓	✓
	Software tool-supported telemetry access	✓	-	-	-	-
	Remote diagnosis and maintenance by robot manufacturer	✓	✓	✓	✓	✓

Interaction with physical and digital world (incl. other CPS) by means of actuators acting on physical processes, via system interfaces, via human-machine interfaces	Collaboration features for interaction between two or more robots, e.g. motion coordination of robots	✓	✓	✓	✓	✓
	Collision detection features	✓	✓	✓	✓	✓
	Collision avoidance strategies for human robot cooperation	✓	✓	✓	✓	✓

Table 2. Industrial robots as CPS in the sense of opened and linked-up systems in contrast to embedded systems

3.3 Industrial robots as CPS in the sense of industry platform constructs

We identified only one area where steps towards an industry platform with a platform leader and complementors as constitutive agents have been made. ABB has recently (Q4 2014) made a website available, where complementors can publish their add-ons developed for ABB's development environment and robots. Thus, ABB as platform leader enables interaction between two independently acting groups (users and complementors) via a platform. Network effects are central to this business model, that is, the more users from one group use the platform, the more attractive it becomes to users from the other group, and vice versa (Gassmann et al. 2014, ABB Robot Apps).

All robots have communication interfaces that are used to communicate with external periphery. These interfaces are essential to create todays' robot cells and workstations and are always open or at least partially open. System integrators use these interfaces to integrate robots into cells or workstations. On the contrary, internal communication interfaces developed by the robot manufacturers are closed to users, system integrators or potential complementors. Only opening these interfaces can enable innovative, complementary products or services, and the use of industrial robots in innovative ways. An exception here is again ABB. In the case of ABB, a transition to open internal interfaces can be observed. In its newly released control system (ABB RobotWare 6, released Q4 2014), ABB allows an unprecedented access to its internal control system functions.

ABB visibly made the first step to transition towards an industry platform model and potentially unlimited pool of external capabilities of innovation. Interfacing the ABB controller is possible by 3rd party equipment, due to the fact that the internal functions of the controller have been made available through web services. Creating new software components in form of add-ons for the ABB controller and development environment is encouraged by the possibility to publish these components on a website. However, the options to create and publish software components are limited due to the fact that ABB's SDK indeed provides access to internal functions of the robot, but the user or developer cannot replace or modify an internal software component or function by a custom-made one. Furthermore, the marketplace for the 3rd party software components only allows free exchange of components. We could not observe similar tendencies at the other four analysed industrial robot manufacturers. The support of G-Code—a programming language for machine tools—by Kuka industrial robots is a step towards openness, since G-Code is a standardized programming language. However, G-Code together with a specialized trajectory planning software package is dedicated for machining operations with robots—a niche market with 1% of worldwide operational robot stock at this point in time (IFR Statistical Department 2013). ABB offers robots for machining applications as well, however without G-Code compatibility. Finally, none of the manufacturers permits access to the underlying operating system (Kuka.CNC, ABB RobotStudio Machining PowerPac, ABB Robot Web Services, ABB WebWare Server, Yaskawa FS100 Controller).

We could not identify a truly modular software architecture that is extensible by third parties with the top five industrial robot manufacturers. Again, just the software architecture of ABB's industrial robots exhibits first steps towards migrating to a modularized architecture: In a niche application area, ABB has introduced a tool that they market as a software platform. Spot welding requires precise control and the aim of ABB is to allow spot welding equipment manufacturers to create their own software packages for controlling the spot welding equipment carried by ABB robots, based on a software framework that ABB maintains. Software package in this case refers to a configuration package; the available instruction set is not expanded. Nevertheless, it is still a remarkable initiative (ABB RobotWare - Spot IRC5).

Table 3 outlines the results of our analysis from the perspective of this chapter.

CPS' abilities and characteristics in general	**Current implementation in the industrial robot domain**	**ABB**	**Kuka**	**Kawa-saki**	**Yas-kawa**	**Fanuc**
Platform leader and complementors as constitutive agents	Platform for publishing 3rd party add-ons	✓ / -	-	-	-	-
Network affected ecosystem governance	Platform for publishing 3rd party add-ons	✓ / -	-	-	-	-
Open interfaces and thus potentially unlimited pool of accessible innovative capabilities	Communication interfaces to external periphery - open or partially open	✓	✓	✓	✓	✓
	Internal communication interfaces - open or partially open	✓ / -	-	-	-	-
	Standardized programming language for niche applications	-	✓	-	-	-
	Access to the underlying operating system / real-time data	-	-	-	-	-
Modular design with core and periphery	Modular software architecture	✓ / -	-	-	-	-

Table 3. Industrial robots as CPS in the sense of industry platform constructs

4 DISCUSSION

4.1 State of the practice

Industrial robots potentially possess all necessary abilities and characteristics of CPS according to our framework of analysis introduced in section 2. As to the two perspectives of subsections 2.1/3.1 and 2.2/3.2, we have identified considerable, current implementations in the industrial robot domain. However, concerning the perspective on CPS as industry platform constructs (subsections 2.3/3.3), takes the industrial robotics branch only certain small steps towards industry platforms. In the case of ABB, a trend towards migrating to an industry platform can be clearly observed, but the implementation is not complete. In case of Kuka, the vision exists (Kuka 2014), but necessary approaches are not visible. The other three industrial robot manufacturers examined currently lack considerations or implementations. Their practices clearly point to supply chain platforms.

Both platform conceptualizations are—besides the internal platform—parts of Gawer's (2014) integrative framework upon which we build our perspective on CPS as industry platform constructs. The framework classifies technological platforms within three increasingly broader organizational settings: Within firms, across supply-chains, and within ecosystems. Gawer assigns a corresponding type of platform to each of the three organizational settings: Internal platform (within firms), supply chain platform (across supply chains), and industry platform (within ecosystems). The platform types differ in their level of analysis, their constitutive agents and technological architecture, the nature of their interfaces, their innovative capabilities and their coordination mechanism.

All examined manufacturers except for ABB offer add-ons within the context of supply chain platforms. They develop these components by themselves or have contractual relations to corresponding suppliers. Only ABB has recently enabled also 3rd parties to develop and publish add-ons for its development environment and for its robots. In order to improve dissemination, ABB has introduced a marketplace for 3rd party components—which is clearly a bold step towards an industry platform approach. It assures both the role of ABB as a platform leader and encourages 3rd parties to publish components and so to become complementors in ABB's platform ecosystem.

Industry platforms have open technological interfaces. There are variations within the spectrum of how open these interfaces are, but interface specifications are generally shared with (potential) complementors. In contrast, interfaces of supply-chain platforms are only selectively open—i.e., specifications are shared exclusively across the supply chain (Gawer 2014). In this regard as well, the industrial robot domain is dominated by supply chain models. It is not as far as other industries when we take a close look at the nature or openness of interfaces and connectivity options. There is also not a standardized approach to programming—a problem known and much debated in robotics:

- All manufacturers offer own proprietary programming languages for programing their robot controllers. This aspect, however, would not prohibit an industry platform approach per se, since at the same time all manufacturers offer interoperable communication and data exchange via the OPC UA protocol. OPC UA is open, and as such it should contribute to openness of an industry platform—so it basically allows access to telemetry data, among others. Unfortunately, it has a major drawback: It is not capable of handling real-time data;
- Internal communication interfaces to the robot are closed. Only ABB has made steps towards opening up its controller architecture by allowing access to internal functions, but exchange or altering of these functions is not permitted;
- The option to access the underlying real-time operating system—provided by suppliers within the manufacturers' supply-chain platforms—is not offered by any of the manufacturers. This possibility would ease accessing real-time data on the controller and would enable deploying complex software components on the controller itself;
- The ability of remote maintenance is offered by all manufacturers and it is a powerful tool, but the option to execute such remote maintenance operations remains limited to the manufacturer himself or to specialized service providers within the supply-chain platform.

4.2 Potentials

In the context of platforms, opening a system to complementary development positively affects innovation by drawing on a wider set of accessible external capabilities and distributed heterogeneous knowledge (Chesbrough 2003), as well as independent experimentation (Gawer 2014). In this sense, the innovation power of an industry platform cannot be compared to the innovation power of a supply-chain platform—the type of platform that dominates the industrial robotics domain. An interesting specificity of industry platforms is that the platform leader does not need to know ex-ante who or where innovators might be. Potential innovators of complementary products identify themselves to the platform's ecosystem. In fact, for truly successful industry platforms, the end use of the product or service does not seem to be fully predetermined by the platform owner. This nature of relationship between the platform leader and his (potential) complementors creates unprecedented scope for innovation on complementary products, services, and technologies (Gawer 2014; Gawer and Cusumano 2014). Thus, in this regard, research also refers to the (industrial) platform's ecosystem as the platform's "innovation ecosystem" (Adner and Kapoor 2010) or as "ecologies of complex innovation" (Dougherty and Dunne 2011). Nambisan and Sawhney (2011) refer to a "shift from firm-centric innovation to network-centric innovation".

Exploiting the full potential of industrial robots understood as CPS according to our framework of analysis could allow broadening of the application areas of industrial robots. The lack of innovative applications is a problem of industrial robots. For more than 20 years, industrial robots have been limited to roughly the same set of tasks—welding, material handling, dispensing and other repetitive and high volume tasks—but novel applications are missing (ROS-Industrial 2015). Current research addresses many novel technologies—not necessarily from the industrial robotics domain—that, on the one hand, are promising because they might improve how we currently use and interact with robots. On the other hand, combining these technologies with the current practice in industrial robotics constitutes serious challenges.

Uncertainty sensing is important for robotics and also automations (Kehoe et al. 2015, Glover et al. 2008) but it requires high computational power. A similar problem exists in the case of object recognition (Kehoe et al. 2015, Van den Berg et al. 2011). In tomorrow's factory, the way we interact with robots will completely change. Voice and gesture recognition technologies are promising new possibilities, which do not fit easily in the motion control-focused control architectures of today's industrial robots (IFR Statistical Department 2013). Machine learning applications face similar problems. Multi-robot cooperation in a flexible way is also an issue intensely researched, which is hard to apply in an industrial setting with the currently available industrial robot controllers of the top five manufacturers (Capitan et al. 2013). Integration of all these technologies into industrial robots—many more examples could be listed here—is neither likely to be done by the industrial robot manufacturer alone nor within the organizational setting of a supply-chain platform, where accessible innovative capabilities are limited to the supply-chain's capabilities. In an industry platform approach, the robot manufacturer would take the role of a platform leader and would be responsible for allowing and encouraging implementations of such technologies and applications by its innovation ecosystem.

Besides product-driven innovations, understanding industrial robots as CPS according to our framework of analysis—i.e. as CPS in the sense of industry platform constructs, among others—opens up considerable potentials for innovative business models. Platform leaders must strive to establish a set of business relationships that are mutually beneficial for their ecosystem participants. They must be able to articulate a set of mutually enhancing business models (Gawer and Cusumano 2014). Mikusz et al. (2015) identify 16 business model patterns that are capable of transforming product innovations, due to abilities and characteristics of CPS and the underlying technical platforms, into business model innovations. Further, the authors analyze, whether and to what extent these patterns solely or in combination are in use in the automotive industry relating to the "connected car", understood as a vehicular CPS. Based on our analysis in section 3, the following subset of the abovementioned business model patterns seem especially interesting to the industrial robot domain.

In the Add-On business model pattern, a competitively priced core offering can be upgraded by numerous extras. One can interpret the Add-On pattern in the sense of a Digital Add-On, which refers to the possibility to upgrade a physical asset by complementary digital services or features—brought out by complementors (Gassmann et al. 2014, Fleisch et al. 2014). A simple use-case for industrial robots could be, for example, a 3rd party software component that, instead of using the robot's internal trajectory planning functions, exchanges these and plans trajectories with sloshing suppression for liquid handling. Our analysis shows that the current implementation of CPS' abilities and characteristics in the industrial robot domain enables the Add-On pattern only to a certain extent. Currently, only ABB goes as far as offering an SDK and the opportunity for 3rd parties to implement components. ABB's modular software architecture is also a key to the Add-on pattern. Accessing the underlying operating system and the robot controller's internal functions via an API would enable implementation of more complex software add-ons. Finally, currently only ABB has introduced a marketplace for 3rd party components, but offers no monetization options to complementors.

Based on the Internet of Things (IoT), Fleisch et al. (2014) introduce the Product as Point of Sales business model pattern. It refers to a physical product, which becomes site of (digital) sales and services potentially brought out by complementors, that is, a Product as Point of Sales (Fleisch et al. 2014). A use-case for industrial robots could be, for example, a robot that autonomously offers the opportunity to the robot operator or supervisor to order a new milling tool (brought out by a complementor), just in time before the lifetime is exceeded. The necessary access to internal and external sensors for this use-case is already given.

The Object Self-Service business model pattern refers to the ability of smart things—here industrial robots—to independently place orders on the internet (Fleisch et al. 2014). This pattern is similar to the previous one with one exception: The robot decides completely autonomously about placing an order, i.e. without interaction with the robot operator or supervisor. The additional decision making capability of the robot can itself be considered as an add-on.

In the Lock-In business model pattern, customers are literally 'locked-in' to a provider's ecosystem. Shifting to another provider's ecosystem would imply considerable switching costs. A Lock-In is either generated by means of technological mechanisms or substantial interdependencies of products or services (Gassmann et al. 2014). Based on the IoT, Fleisch et al. interpret Lock-In as a Digital Lock-In in physical products, which refers to a sensor-based, digital initialization of data that limits compatibility, prevents counterfeits and ensures warranties (Fleisch et al. 2014). In industrial robotics, we already identified a pre-existing lock-in pattern: All five manufacturers offer their own proprietary programming language; some offer GPL but with proprietary SDKs. A challenge here is to find the right degree of lock-in by combining open interfaces (also pre-existent) with a not necessarily only technical lock-in effect.

4.3 Future research directions

On the one hand, the state of the practice shows that the industrial robotics domain is in essence dominated by supply chain models. Only ABB visibly made first steps to a transition towards an industry platform approach, but the implementation is not complete. On the other hand, the previous section shows the offered potentials in industrial robotics by migrating to an industry platform. First and foremost, the innovation power of an industry platform exceeds the innovation power of a supply-chain platform by far. Exploiting the full potential of industrial robots understood as CPS according to our framework of analysis—i.e. as CPS in the sense of industry platform constructs, among others—enables innovative, complementary products or services, and the use of industrial robots in innovative ways. It also opens up considerable potentials for innovative business models. We outlined a set of business model patterns that can transform product innovations, enabled by abilities and characteristics of CPS, into business model innovations in the industrial robot domain.

Further research is needed to explain this contradictory finding—questions concerning a CPS platform approach to industrial robots remain: Why has only one of the five key players moved towards an industry platform approach? What is the right degree of openness within the range between the supply-chain and the industry platform, whereby there are also variations of how open technological interfaces of industry platforms are. Recent research on platforms highlights the complex trade-off between open & closed and the complex interplay between platform openness, innovation, and competition (Boudreau 2010, Boudreau 2012, Eisenmann et al. 2009, Gawer 2009, Gawer and Cusumano 2014, Greenstein 2009, Schilling 2009). Drawing on this research stream in order to address the abovementioned research directions seems promising.

Interfaces around an industry platform should be sufficiently open to attract complementors into the platform ecosystem—this resonates well with research by Chesbrough (2003) and Baldwin and von Hippel (2005) on open innovation. The degree of openness has an influence on the extent to which facilitation of innovation can happen (Gawer and Cusumano 2014). However, at the same time industry platforms operate in ways that combine innovation with increased competitive tensions within their ecosystems, as well as across ecosystems—i.e. competitive tension between platforms, between complementors, and between the platform owner and its complementors. For example, while a large proportion of the platform ecosystem's complementors will innovate in ways that are complementary to the platform, a number of them will start innovating in ways that become competitive to the platform (Gawer 2014). We have observed a trend in industrial robotics that is very interesting from the point of view of the interplay between platform openness, innovation, and competition: Instead of requiring that their industrial robots were to be programmed in their own programming environments, Yaskawa and Kuka allow other industrial automation equipment manufacturers to use their own proprietary automation controller software also to control robots. Is this the right degree of openness?

Another future research direction regarding the issues raised could be to apply the service system abstraction to industrial robots, understood as CPS according to our framework of analysis. Mikusz (2015) proposes a conceptual framework for CPS from the Service-Dominant logic (S-D logic) perspective. It aims to provide insights into service architectures of CPS (here industrial robots).

Service architecture transforms the value proposition of a service system into a configuration of actors, resources, and activities of value co-creation. The framework considers CPS as complex service systems in the sense of Service Science and S-D logic—i.e. as dynamic value co-creation configurations of resources, including people, organizations, shared information, and technology, all connected internally and externally to other service systems by value propositions (Maglio et al. 2009). Service Science's and S-D logic's systems orientation fits very well with main CPS characteristics—CPS' capability to temporarily build systems of systems with dynamically changing system boundaries, and their characterization as globally interconnected, cooperative systems with distributed and changing control, among others—both can be well represented by atomic and composite service systems. The framework assumes that value can only be generated by complex service systems that are assembled from different service contexts. In other words, value can only be generated by novel and ad-hoc composed value propositions of local service providers, electronic service providers, as well as other CPS—all transmitted by certain defined interaction channels.

5 LIMITATIONS

Our results are limited by its exploratory nature and need further elaboration, scrutiny, and competing views. Further empirical work, now beginning, is needed and is to go far beyond our first exploratory study. It should provide a deeper examination of both technical and economic challenges and advantages of a CPS platform approach to industrial robots.

References

ABB Maintenance. http://www.abb.de/product/seitp327/decfefae18d4a15ac1257a5b00446908.aspx?_ga=1.214815319.1154918307.1422606807 (accessed March 18, 2015).

ABB Robot Application Builder. http://new.abb.com/products/robotics/application-software/robot-application-builder (accessed March 18, 2015).

ABB Robot Apps. http://www.abb.com/product/ap/seitp327/5dd4fbc7afcb82d2c12579ad0054401d.aspx (accessed March 18, 2015).

ABB RobotStudio Machining PowerPac. http://new.abb.com/products/robotics/application-software/machining (accessed March 18, 2015).

ABB RobotWare - Spot IRC5. http://www.abb.com/product/seitp327/b91afba78aad1462c12570c900402145.aspx# (accessed March 18, 2015).

ABB Robot Web Services. http://developercenter.robotstudio.com/Index.aspx?DevCenter=Robot_Web_Services (accessed March 18, 2015).

ABB SDK/RobotStudio Add-Ons. http://developercenter.robotstudio.com/Index.aspx?DevCenter=RobotStudio&OpenDocument&Url=html%2f130a697c-c8ad-409d-8300-0c0ede835b95.htm (accessed March 18, 2015).

ABB WebWare Server. http://new.abb.com/products/robotics/application-software/webware-server (accessed March 18, 2015).

Adner, R. and Kapoor, R. (2010). Value creation in innovation ecosystems: how the structure of technological interdependence affects firm performance in new technology generations. Strategic Management Journal, 31 (3), 306–333.

Baldwin, C.Y. and von Hippel, E. (2011). Modeling a paradigm shift: from producer innovation to user and open collaborative innovation. Organization Science, 22 (6), 1399–1417.

Boudreau, K.J. (2010). Open platform strategies and innovation: granting access vs. devolving control. Management Science, 56 (10), 1849–1872.

Boudreau, K.J. (2012). Let a thousand flowers bloom? An early look at large numbers of software app developers and patterns of innovation. Organization Science 23 (5), 1409–1427.

Broy, M., Cengarle, M. V. and Geisberger, E. (2012). Cyber-Physical Systems: Imminent Challenges. In Large-Scale Complex IT Systems - 17th Monterey Workshop 2012 (7539) (Calinescu, R. and Garlan, D. Eds.), 1-28, LNCS, Springer, Oxford.

Buxmann, P., Diefenbach, H. and Hess, T. (2013). The Software Industry: Economic Principles, Strategies, Perspectives. Springer, Berlin and Heidelberg.

Capitan, J., Spaan, M.T.J., Merino, L. and Ollero, A. (2013). Decentralized multi-robot cooperation with auctioned POMDPs. The International Journal of Robotics Research, 32 (6), 650-671.

Chesbrough, H.W. (2003). Open Innovation: The New Imperative for Creating and Profiting from Technology. Harvard Business School Press, Boston, MA.

Chen, Y., Du, Z. and Garcia-Acosta, M. (2010). Robots as a Service in Cloud Computing. In Proceedings of the fifth IEEE International Symposium on Service Oriented System Engineering, (O'Conner, L. Ed.), 151-158, Nanjing, China.

Chen, Y. and Hu, H. (2013). Internet of intelligent things and robot as a service. Simulation Modelling Practice and Theory, 34, 159-171.

Dougherty, D. and Dunne, D. (2011). Organizing ecologies of complex innovation. Organization Science, 22 (5), 1214–1223.

Eisenmann, T., Parker, G. and Van Alstyne, M. (2009). Opening platforms: how, when, and why? In Gawer, A. (2009), 131–162.

Fanuc Controller R-30iB. http://www.fanucrobotics.se/en/products/controllers/r-30ib (accessed March 18, 2015).

Fanuc Coordinated Motion Package. http://www.fanucrobotics.de/de/products/software/controller%20 software (accessed March 18, 2015).

Fanuc Remote Diagnostics. http://www.fanucrobotics.co.uk/~/media/Global/Files/Downloads/ Magazines/After%20Sales/EN/Remote_Diagnos_EN.ashx (accessed March 18, 2015).

Fleisch, E., Weinberger, M. and Wortmann, F. (2014). Business Models and the Internet of Things. Bosch IoT Lab White Paper, August 2014, St. Gallen.

Gassmann, O., Frankenberger, K. and Csik, M. (2014). The Business Model Navigator: 55 Models that will Revolutionise your Business. Pearson, Harlow, UK, New York.

Gawer, A. (2009). Platforms, markets and innovation. Edward Elgar, Cheltenham, UK.

Gawer, A. (2014). Bridging differing perspectives on technological platforms: Toward an integrative framework. Research Policy, 43, 1239-1249.

Gawer, A. and Cusumano, M. A. (2014). Industry Platforms and Ecosystem Innovation. Journal of Product Innovation Management, 31 (3), 417-433.

Geisberger, E. and Broy, M. (2015). Living in a network world: Integrated research agenda Cyber-Physical Systems. Acatech – National Academy of Science and Engineering, March 2015

Glover, J., Rus, D. and Roy, N. (2008). Probabilistic Models of Object Geometry for Grasp Planning. In Proceedings of Robotics: Science and Systems IV, Zurich, Switzerland, 278-285.

Greenstein, S. (2009). Open platform development and the commercial internet. In Gawer, A. (2009), 219–48.

IFR Statistical Department (2013). World Robotics. Industrial Robots 2013.

ISO (2011). 10218-1:2011, Robots and robotic devices – Safety requirements for industrial robots – Part 1: Robots.

ISO (2012). ISO 8373:2012, Robots and robotic devices – Vocabulary.

Kagermann, H., Wahlster, W. and Helbig, J. (2013). Securing the future of German manufacturing industry: Recommendations for implementing the strategic initiative INDUSTRIE 4.0. Acatech – National Academy of Science and Engineering, April 2013.

Kagermann, H., Riemensperger, F., Hoke, D., Helbig, J., Stocksmeier, D., Wahlster, W. and Scheer, A. (2014). Recommendations for the Strategic Initiative Web-based Services for Businesses. Acatech – National Academy of Science and Engineering, February 24.

Kawasaki E-Controller. http://www.kawasakirobotics.com/userAssets/productFiles/E-Controller.pdf (accessed March 18, 2015).

Kehoe, B., Patil, S., Abbeel, P. and Goldberg, K. (2015). A Survey of Research on Cloud Robotics and Automation. IEEE Transactions on Automation Science and Engineering, 12 (2).

Kuka (2014). Annual Report 2013. February 26.
Kuka.CNC. http://www.kuka-robotics.com/germany/de/products/software/application_software (accessed March 18, 2015).
Kuka Control & Observe. http://www.kuka-robotics.com/germany/de/products/software/control _observe (accessed March 18, 2015).
Kuka Conveyortech. http://www.kuka-robotics.com/germany/de/products/software/application_ software/ (accessed March 18, 2015).
Kuka.HMI. http://www.kuka-robotics.com/germany/de/products/software/additional_functions (accessed March 18, 2015).
Kuka.PLC mxA. http://www.kuka-robotics.com/res/sps/a737ee03-5832-4c95-9d91-84e0de80c664_ KUKA_mxAUTOMATION_DEUTSCH.pdf (accessed March 18, 2015).
Kuka Remote Control. http://www.kuka-robotics.com/germany/de/products/software/remotecontrol/ (accessed March 18, 2015).
Lee, E.A. (2006). Cyber-Physical Systems – Are Computing Foundations Adequate. Position Paper for NSF Workshop on Cyber-Physical Systems, Austin, USA, 1-9.
Lee, E.A. (2008). Cyber Physical Systems: Design Challenges. In 11th IEEE Symposium on Object Oriented Real-Time Distributed Computing, Orlando, USA, 363-369.
Maglio, P., Vargo, S.L., Caswell, N. and Spohrer, J. (2009). The service system is the basic abstraction of service science. Information System and e-Business Management Journal 7, 395–406.
Mikusz, M., Jud, C. and Schäfer, T. (2015). Business Model Patterns for the Connected Car and the Example of Data Orchestrator. In Proceedings of the 6th International Conference on Software Business, (Fernandes, J.M. et al. Eds.), ICSOB 2015, Springer LNBIP 210, 167-173.
Mikusz, M. (2015). Towards a Conceptual Framework for Cyber-Physical Systems from the Service-Dominant Logic Perspective. In Proceedings of the Twenty-first Americas Conference on Information Systems (AMCIS), AIS electronic library, Puerto Rico (forthcoming).
Nambisan, S. and Sawhney, M. (2011). Orchestration processes in network-centric innovation: evidence from the field. Academy of Management Perspectives, 25 (3), 40–57.
Pussep, A., Schief, M. and Widjaja, T. (2012). The Software Value Chain: Methods for Construction and Their Application. In Proceedings of the 20th European Conf. on Inf. Sys., Barcelona.
Rajkumar, R., Lee, I., Sha, L. and Stankovic, J. (2010). Cyber-physical systems: the next computing revolution. In Proceedings of the 47th Design Automation Conference. ACM, 731-736, New York.
ROS-Industrial (2015). The Challenge: Transitioning Robotics R&D to the Factory Floor. http://rosindustrial.org/the-challenge/ (accessed May 20, 2015).
Schief, M. and Buxmann, P. (2012). Business Models in the Software Industry. In 45th Hawaii International Conference on System Sciences, 3328-3337.
Schilling, M. A. (2009). Protecting or diffusing a technology platform: Trade offs in appropriability, network externalities, and architectural control. In Gawer, A. (2009), 192–218.
Statista (2015). Leading companies in the global industrial robot market in 2011, based on market share. http://www.statista.com/statistics/257177/global-industrial-robot-market-share-by-company/ (accessed March 18, 2015).
Van den Berg, J, Abbeel, P. and Goldberg, K. (2011). LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information. International Journal of Robotics Research, 30 (7), 895-913.
Yaskawa Advanced PP Customization SDK. http://www.motoman.de/de/produkte/software/product-view/?no_cache=1&tx_catalogbasic_pi1%5Buid%5D=976&cHash=14e8b16b19f7f9a6a2ce856877717f41 (accessed March 18, 2015).
Yaskawa FS100 Controller. http://www.motoman.com/products/controllers/fs100_controller.php (accessed March 18, 2015).
Yaskawa Program Motoman Robots with Rockwell Automation PACs. http://www.motoman.com/products/controllers/mlx200_controller.php (accessed March 18, 2015).
Yaskawa. Total Customer Support Overview: Service, Maintenance, Spare Parts, Repair, Training. http://docs.yaskawa.eu/Brochure%20tcs%20YEUK/index.html#1 (accessed March 18, 2015).