International Journal of Production Research

ISSN: 0020-7543 (Print) 1366-588X (Online) Journal homepage: https://www.tandfonline.com/loi/tprs20

Feature-based function block control framework for manufacturing equipment in cloud environments

Göran Adamson, Lihui Wang & Philip Moore

To cite this article: Göran Adamson, Lihui Wang & Philip Moore (2019) Feature-based function block control framework for manufacturing equipment in cloud environments, International Journal of Production Research, 57:12, 3954-3974, DOI: 10.1080/00207543.2018.1542178

To link to this article: https://doi.org/10.1080/00207543.2018.1542178

Published online: 27 Nov 2018.

International Journal of Production Research, 2019
Vol. 57, No. 12, 3954–3974, https://doi.org/10.1080/00207543.2018.1542178

Feature-based function block control framework for manufacturing equipment in cloud environments

Göran Adamson [a*], Lihui Wang [a,b] and Philip Moore[a]

[a]*Virtual Systems Research Centre, University of Skövde, Sweden;* [b]*Department of Production Engineering, Royal Institute of Technology, Stockholm, Sweden*

(*Received 2 February 2018; accepted 23 October 2018*)

The ability to adaptively control manufacturing equipment in cloud environments is becoming increasingly more important. Industry 4.0, supported by Cyber Physical Systems and the concept of on-demand, scalable and pay-for-usage resource-sharing in cloud environments offers many promises regarding effective and flexible manufacturing. For implementing the concept of manufacturing services in a cloud environment, a cloud control approach for the sharing and control of networked manufacturing resources is required. This paper presents a cloud service-based control approach which has a product perspective and builds on the combination of event-driven IEC 61499 Function Blocks and product manufacturing features. Distributed control is realised through the use of a networked control structure of such Function Blocks as decision modules, enabling an adaptive run-time behaviour. The control approach has been developed and implemented as prototype systems for both local and distributed manufacturing scenarios, in both real and virtual applications. An application scenario is presented to demonstrate the applicability of the control approach. In this scenario, Assembly Feature-Function Blocks for adaptive control of robotic assembly tasks have been used.

Keywords: manufacturing feature; adaptive control; cloud manufacturing

1. Introduction

Surviving in an increasing globalisation, manufacturing companies are focusing on adopting more cost-effective manufacturing systems to remain competitive (Kusiak 2017; Valilai and Houshmand 2013). To be able to be competitive on a global marketplace, collaboration within global supply chains and manufacturing networks for critical and complex manufacturing activities such as design and manufacturing is of high interest for many companies. Sharing resources, knowledge and information between geographically distributed manufacturing entities can make them more agile and cost-effective, with higher resource' utilisation, leading to a competitive edge, in a win-win scenario for all participants (Ding, Yu, and Sun 2012).

Recently, research covering the collaboration and resource-sharing in all parts of the product development life-cycle has shown a growing interest. With new opportunities arising from improvements within modern information and communication technology, service and information-driven manufacturing has become a focused research topic and already made some progress within collaborative and distributed manufacturing (Li, Zhang, and Chai 2010). Cloud Manufacturing (CM) is evolving as a new manufacturing paradigm to match this trend, enabling the mutually advantageous sharing of resources, knowledge and information between distributed companies and manufacturing units. The concept of offering computer resources as services can be adopted in manufacturing, with manufacturing resources being offered as different services, i.e. Design-as-a-Service (DaaS), Machining-as-a-Service (MCaaS), Assembly-as-a-Service, etc. (Xu 2012). This have led to new and challenging requirements regarding the coordinated planning, control and execution of discrete manufacturing operations in collaborative and networked environments.

This shift in manufacturing orientation also increases the complexity of realising adaptive control for such distributed real-time environments dramatically (Meier, Seidelmann, and Mezgár 2010). The level of complexity will become significantly higher, as the nature of a distributed manufacturing environment presents a higher degree of uncertainties (Adamson et al. 2015). Variations and unforeseen events may be inflicted by all participating companies' internal and external variations within collaborative manufacturing missions. The ability to handle such influences of uncertainty requires both flexibility and adaptability to be competitive (Boutellier, Gassmann, and von Zedtwitz 2008). Therefore, a prominent property for an

*Corresponding author. Email goran.adamson@his.se

adaptive and distributed control structure is the dynamic coordination and distribution of decision-making to both global and local environment instances (Monostori et al. 2010). This would enable adaptive system control as adjustments to any changes, not least for shop-floor run-time variations. Traditional planning and control systems are often not able to handle unforeseen changes efficiently (Xu, Wang, and Newman 2011). Therefore, planning, scheduling and control of physical manufacturing equipment in distributed environments will be crucial for the successful realisation of CM (Adamson et al. 2015).

This paper introduces a novel method for adaptive manufacturing equipment control in cloud environments, presenting the construct of a feature-based Function Block control concept which encompasses the complete manufacturing control structure, from supervisory control on a cloud level, down to local generation and execution of equipment control at the shop floor and machine controller levels.

The presented research is an extension of the authors' previous research within feature-based Function Block control, of which some work is also referenced in this paper.

2. IEC 61499 event-driven function blocks

The ability to efficiently adjust to changing conditions, adaptability, is an important property of a manufacturing control system (Groover 2016). To successfully handle unpredictable events negatively affecting the performance of a manufacturing system requires both adaptive and distributed run-time decision-making, but also effective execution of these decisions. One approach to limit the negative influence of uncertainty and unpredictable behaviour on manufacturing performance is to use real-time manufacturing information (Wang et al. 2012). Using real-time system information for both planning and control of a manufacturing system means that the time span between decision-making and actual execution can be narrowed down to a minimum, facilitating more correct decisions as well as decreasing possible negative impact of uncertainty. Using actual events within a distributed control system to trigger the dynamic generation of the required control activities would make possible adaptive decision-making and dynamic control capabilities, as an important and valuable built-in control system property to handle uncertainty.

The concept of event-driven Function Blocks supports this approach, as it enables the use of online information for dynamic and distributed decision-making, as well as dynamic control capabilities that are able to handle, in a responsive and adaptive way, different kinds of uncertainty. Applying such Function Blocks for the control of manufacturing equipment implies giving the control system more intelligence and autonomy to better handle and adapt to changes, for a more successful fulfilment of the manufacturing objectives (Monostori et al. 2010).

2.1. Introduction to IEC 61499

Event-driven Function Blocks are initially defined in the IEC 61499 standard (IEC 2005), which explains the usage, development and implementation of Function Blocks in distributed industrial process measurement and control systems, in a component-oriented approach. The standard describes a generic modelling approach for distributed control applications enabling interoperability, re-configurability and portability for distributed control systems, facilitated through event-driven Function Blocks. The primary purpose of IEC 61499 is not that of a programming methodology, but instead it describes a system architecture, and provides a set of models to describe distributed control systems using event-driven Function Blocks in a real-time execution environment (Lewis 2001; Vyatkin 2011; Zoitl 2008).

The standard supports intelligence to be decentralised and wrapped in software components, which can be distributed in a system control network. An event-driven Function Block-based control system can, therefore, be applied to control various industrial systems as well as be used for high-level process planning. The control approach is flexible and versatile as it can be designed to handle both execution control, process monitoring and the scheduling of dynamic resources (Lewis 2001).

The IEC 61499 Function Block is defined as an event-triggered component with inputs and outputs for events as well as data, with algorithms, internal variables controlled by the Execution Control Chart (ECC) (Figure 4). The execution of its algorithms, triggered by arriving input events, determines the Function Block behaviour. Function Block algorithms will read data from incoming input data when executing, and then produce new output data. The completion and availability of the output data will then be announced by output events. The algorithm execution and scheduling are controlled by the ECC, a finite state machine with different states, transitions and actions. Function Blocks are intended to encapsulate a software solution for a dedicated task, using one or several algorithms. As such, they can encapsulate generic functionality which can be used in different control scenarios. By combining Function Blocks into networks, complete control applications with an aggregated higher-level functionality can be realised.

2.2. IEC 61499 applications

Research about the implementation of IEC 61499 Function Blocks has been ongoing for a while, at least since the late 1990s, and a variety of approaches using such Function Blocks have been proposed. It seems that the majority of IEC 61499 applications are limited to low-level process control for PLCs, which are not able to handle issues of uncertainty and adaptivity regarding process planning and execution control for complex machining or robotic operations in high-level manufacturing systems (Wang et al. 2012).

A rather common Function Block application is the system design of autonomous distributed systems with intelligent control components. Early research on using Function Blocks describes holonic control (Wang et al. 2001). Other examples of how IEC 61499 has been studied and discussed in the research literature are: an automatic verification of industrial control systems based on function block technology (Völker and Krämer 2002), the development of an architecture for Function Block-oriented engineering support systems (Thramboulidis and Tranoris 2001) and reconfigurable concurrent Function Block models and their implementations using real-time Java (Brennan et al. 2002).

Research on implementations of IEC 61499 in process control systems are also found (Wang et al. 2012). Real-time execution of IEC 61499 applications, describing the execution elements within a device and different scheduling and implementation approaches is presented by Zoitl et al. (2005), as well as critique against and solutions for, ambiguities concerning execution in the standard, leading to different execution behaviour of elements on different control devices, by Strasser et al. (2011). The development, implementation and use of an IEC 61499 Function Block library for embedded closed-loop control is presented and demonstrated by Strasser, Auinger, and Zoitl (2004) on a real experiment: the control of a challenging seesaw problem. The implementation of a real-time distributed control model using a Java-based platform is introduced by Olsen et al. (2005), where a control application is distributed across two devices, supported by a MANAGER Function Block, able of providing management services for devices.

Targeting the issue of manufacturing equipment control, various applications for e.g. robots and CNC-machines have been described. An open, layered CNC-FB architecture, simplifying the design of CNC machine controllers, is demonstrated by Minhat et al. (2009), Minhat, Xu, and Vyatkin (2009). The architecture is based on STEP-NC (STandard for the Exchange of Product model data – for Numerical Control) (ISO 2007) as the input data model and IEC 61499 as its development platform. The STEP-NC model provides data on the machining operation to be executed. A prototype system with a PC controlled 3-axis CNC vertical milling machine has been used to test the proposed architecture, and to prove that Function Block technology can be used for the development of open and distributed CNC systems. The actual control has been achieved by interfacing the three stepper motors of the machine through the parallel port of the PC, through an EMC controller and through a motor control unit that drives the motors. With this setup, only a signal for direction and a single pulse are needed to move the motors one step in any direction. A Composite-Function Block controls the motors by generating an output sequence on the parallel port.

Targeting the absence of a CNC-controller that is able to directly execute STEP-NC models, an adaptable CNC system based on STEP-NC and Function Blocks was proposed (Wang, Xu, and Tedford 2007). It addresses the issue of porting STEP-NC data to different CNC controllers, to enable a 'Plug-and-Play' functionality. The objectives of this STEP-compliant CNC system with Function Blocks incorporated are: to make product data interchangeable, to enable information flow seamlessly and to have a system that is independent of CAD/CAM systems. In their 'Plug-and-Play' mapping system, a STEP-NC encoder reads data from an STEP-NC supporting system and encodes it into Function Blocks. A Function Block mapping system is then used to translate the STEP-NC code into native CNC-machining G and M codes, which are executed by an executing sub-system. One advantage with this system is that the current CNC machine configurations do not have to be modified. From a controller perspective, machine specific G and M codes can still be used.

An enhanced STEP-NC compliant CNC controller is presented by Huang (2010). In order to adapt the controller to a reconfigurable environment, an extended STEP-NC data model describing machining data from the viewpoint of product family is applied, as well as the Function Block device element model of IEC 61499. The approach is demonstrated on a XY table and linear module in an FMS platform, controlled by a PC with a motion control card. In Doukas, Thramboulidis, and Koveos (2006) an approach applying IEC 61499 Function Blocks for robotic arm motion control is presented, using a PID-based control for issuing motion commands to the robot. The motion behaviour for different variable PID parameters and sampling periods are examined to prove the correctness of the design and the implementation of the control application. Adaptive CNC machining using Function Blocks is presented by Wang and Wang (2018). Function Blocks are embedded in machining processes by combining machining features, representing machining information, e.g. machining sequence, machining parameters, and other relevant machining resources. A reachability-based machining feature sequencing method then generates the machining sequence adaptively to minimise the cutting tool change times. A Function Block based cyber-physical production system for physical human-robot interaction is presented by Yao et al. (2018). The authors use an assembly case to demonstrate the feasibility of the proposed system, for which the IEC 61499 standard is found a suitable technology due to properties such as modularisation, reusability and distributable control. Their results show that the Function Block based Cyber Physical Production System for physical Human-Robot interaction holds the potential capability for secure Human-Robot based assembly. In a literature review (Vyatkin 2011) focusing on how IEC 61499 can enable distributed and intelligent automation, the author concludes that the standard facilitates control systems which may be automatically generated directly from the design documentation using integrated design methodologies. This review also concludes that the standard's benefits for design of control systems, compared to other technologies used for automation control, have been well proven by system integrators with experience of IEC 61499 implementations.

3. Feature-based manufacturing

An effective approach to solving many manufacturing issues is to apply feature-based manufacturing. This approach, which stems from a product perspective, since it builds on the product manufacturing feature concept, is a viable and effective method for adaptive and distributed manufacturing since feature-based manufacturing can be realised through the application of different manufacturing services. By using the concept of manufacturing services, in a similar manner to the use of services within cloud computing, manufacturing resources and capabilities can be provided in distributed environments, e.g. CM, in which device network capabilities, such as Internet of Things, may enable access for controlling distributed manufacturing equipment. Through the use of feature-based and IEC 61499 event-driven Function Blocks (FBs) as smart and distributable decision modules, run-time manufacturing operations in a CM environment may be controlled and executed, in order to meet prevailing manufacturing conditions and requirements. Developed for adaptive and distributed manufacturing equipment control, these modules can be combined into control networks to satisfy different levels of control needs, and ultimately realise the idea of Manufacturing-as-a-Service (MaaS) (Adamson, Wang, and Moore 2017; Herterich, Uebernickel, and Brenner 2015; Wu and Yang 2010).

The use of Manufacturing Features (MfgFs) has many advantages in manufacturing, as it can be applied for different, and cooperative, purposes. Central in this research is a combined approach for use of MfgFs, established from a product, resource, planning and control perspective (Figure 1) (Adamson, Wang, and Moore 2017).

This provides great flexibility as MfgFs are used to:

- detail the product model and the manufacturing task,
- describe generic capabilities of manufacturing resources,
- support and simplify manufacturing planning and control, incl. programming and the run-time generation and execution of control instructions for manufacturing resources.

3.1. Feature information framework

Using MfgFs to describe both products and resources is necessary for their discovery and the matching between manufacturing task requests and available manufacturing resources. For resources, MfgFs for different manufacturing domains are used to describe the resource's ability to complete unique manufacturing operations as product features, through combinations and aggregations of manufacturing resources' functionalities and properties, into manufacturing capabilities (Adamson, Wang, and Moore 2017; Figure 2).

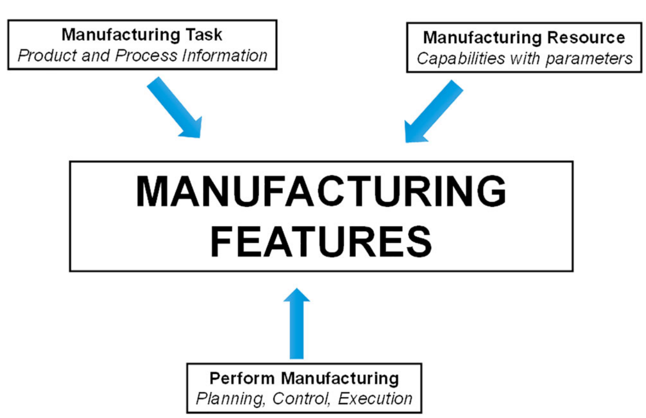

Figure 1. Combined use of Manufacturing Features.

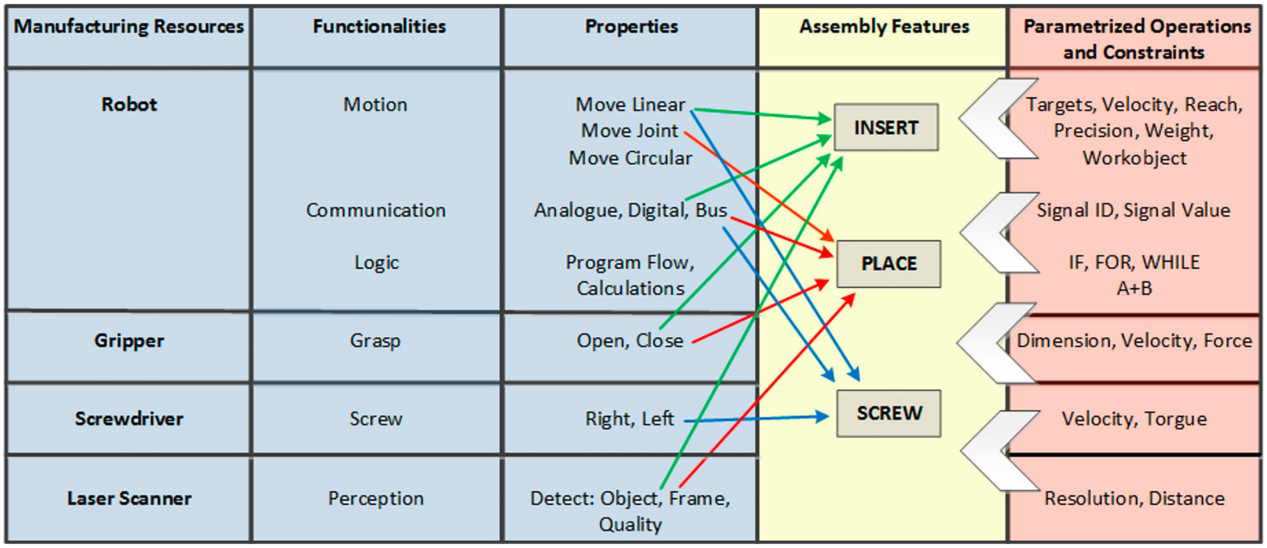

Figure 2. Feature-level assembly resource capability model.

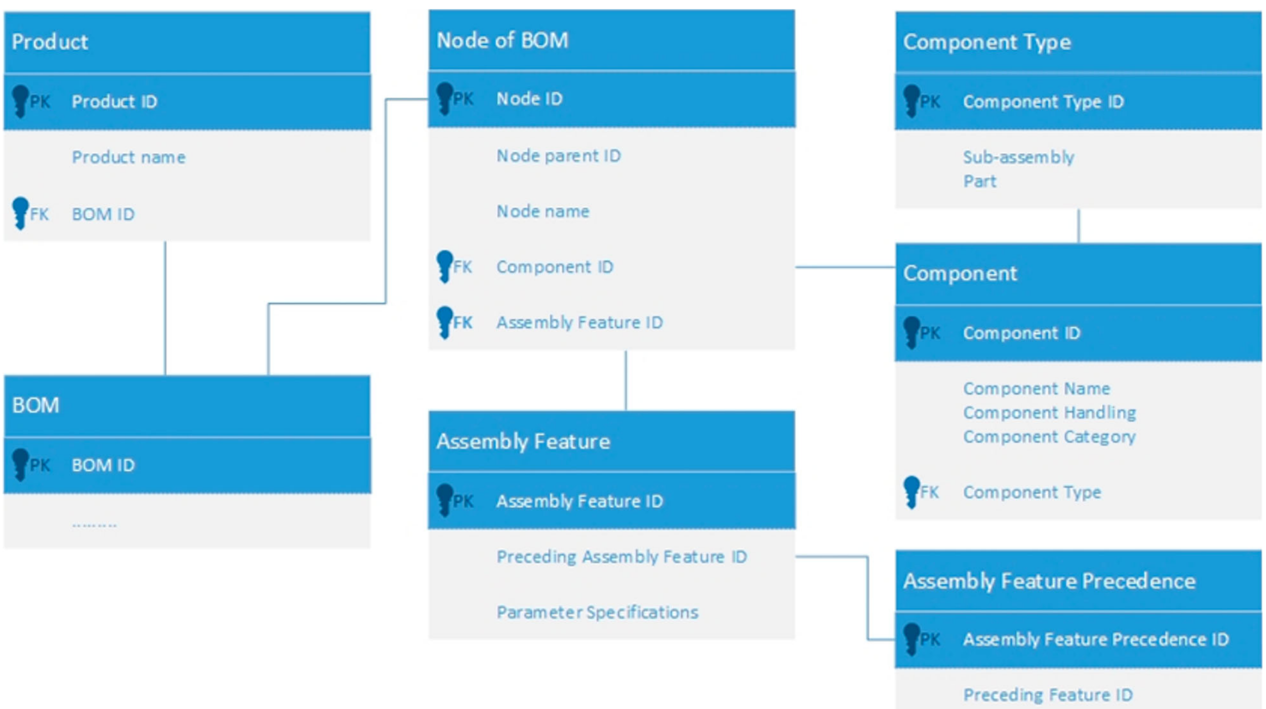

Figure 3. Assembly feature-enriched product data model.

For product manufacturing tasks, MfgFs are used to describe how they are to be manufactured (Adamson, Wang, and Moore 2017; Figure 3).

By combining these descriptions of manufacturing resources capabilities and product manufacturing requirements, a supporting feature Information Framework for discovery and matching of manufacturing resources to products can be realised.

3.2. *Assembly features*

An assembly task defines the work to be performed to assemble something. For a more specific description of the details and use of MfgFs, a robotic assembly task is selected and therefore the concept of Assembly Features (AFs) is described in more detail (Adamson, Wang, and Moore 2017). A robotic assembly task is performed by a robot or a robot cell or station, and may comprise e.g. to assemble two parts to a component/sub-assembly, or to assemble a variety of components into a complete car engine. Here, robotic assembly relates to the manipulation of components (parts and sub-assemblies) for the creation of assembled products, and AFs are used to encode the assembly method between connected components. An assembly feature contains the detailed know-how of how to perform an atomic or low-level assembly operation, and all unique assembly operations can be identified and mapped to AFs. As such, typical basic assembly operations e.g. *Insert*, *Screw* and *Place* are realised as AFs, which are mainly used for coordination of parametrised motions. Except for motions, there are also various actions which need to be performed within assembly tasks, including: signal processing, programme logic, decision-making, and communication. Depending on required functionality, AFs are controlled by various input parameters, e.g. motion modes, target locations, velocities, tolerances, tool IDs, signal IDs and values, etc. (Adamson, Wang, and Moore 2017). In a robotic assembly task, a sequence of different basic assembly operations is necessary to complete the task (Adamson et al. 2015). By grouping different AFs, the coordination and control of a set of robot motions and actions is possible to complete such an assembly task. As product/part design has a primary influence on ease of assembly and associated costs, the Design for Manufacturing and Assembly (DFMA) analysis method can be applied (Selvaraj, Radhakrishnan, and Adithan 2009). This method facilitates the creation of proper design of parts for ease of assembly.

4. Event-driven adaptability using IEC 61499 function blocks and manufacturing features

To practically implement the concept of MfgFs for the adaptive control of manufacturing equipment, an implementation approach for the planning and execution of MfgFs is necessary. For this, an executional mechanism, capable of automatically generating the required equipment control instructions is required. Realising the full potential of IEC 61499 event-driven FB's, the inclusion of MfgFs makes possible an adaptive and flexible control approach for different manufacturing applications. By combining the distributed run-time decision-making properties of event-driven FBs with the manufacturing 'know-how' of MfgFs, it is possible to create the executable manufacturing control system unit Manufacturing Feature-Function Block (MfgF-FB) (Adamson et al. 2015).

The IEC 61499 FB can be regarded as a run-time decision-making module, and its overall behaviour is determined by its ECC which controls the scheduling and execution of the internal algorithms. The functionality of the FB is encapsulated into its algorithms, and arriving input events will trigger these to execute. When executing, the algorithms will read and use available input data, as well as internally available data, for creating dynamically adjusted output data. Output events are used to announce the completion and availability of new output data. Thus, by mapping the desired functional behaviour of MfgFs to the algorithms, this unit provides the encapsulation of manufacturing feature functionality, as well as data transfer, the event-driven process and execution control. When triggered, it will dynamically generate the manufacturing control instructions required to perform such a basic manufacturing operation, e.g. *Insert* for a robotic assembly task (Adamson, Wang, and Moore 2017). The construct of the FB control network must be able to handle time delays prone to appear in distributed environments such as Cloud environments. Therefore, a detailed strategy for the correct and timely reception of input events is necessary. The use of predefined algorithms to handle unexpected situations/events is commonly used for this.

Creating robot assembly functionality includes setting typical robot parameters such as; targets and paths, robot move mode, TCP speed, level of accuracy, tool, reference frame/workobject to use, etc. (Adamson et al. 2015). For successfully performing required robot operations, real-time generation of correct control instructions is necessary, therefore adaptive robot control requires real-time monitoring of the status of the robot and its working environment, and its process prerequisites. The control system's adaptivity property derives from the Assembly Feature-Function Blocks' (AF-FB) ability to adjust their output data to the actual assembly conditions, as the algorithms are triggered in real-time by arriving input events from the robot and its environment (Adamson et al. 2015).

Demonstrating the AF-FB concept, a combination of an event-driven IEC 61499 Basic FB with the AF *Insert*, with its associated ECC, is depicted in Figure 4.

4.1. Combining assembly feature-function blocks for creating assembly control applications

While basic AF-FBs can define the functional relationships between events, data and algorithms for individual AFs fabrication, their combination into an AF-FB network forms a Composite FB representing a high-level assembly task. Figure 5 shows a Composite FB, where the sequence among three parallel AF-FBs is facilitated at runtime by an Event Switch-FB.

4.1.1. Adaptive robot assembly control using assembly feature-function blocks

The construct of the AF-FB control approach provides the ability to dynamically adapt to variations in assembly tasks, such as changes in product designs, changes in assembly component locations, performing operations with a different robot

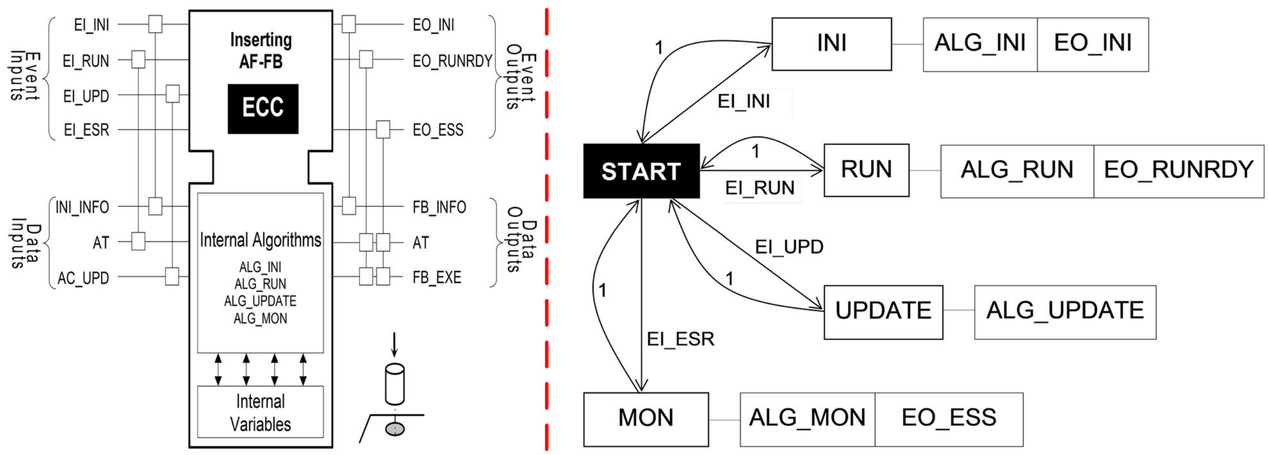

Figure 4. INSERT Assembly Feature-Function Block with ECC.

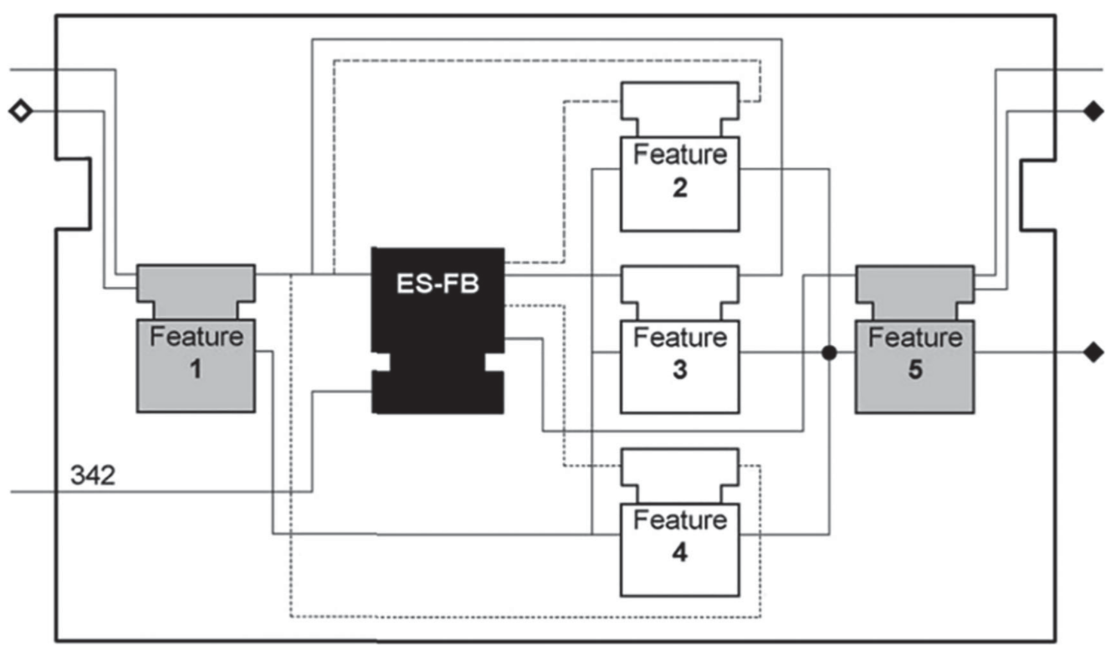

Figure 5. Composite Function Block control application.

tool, one robot completing the operations of another robot, etc. In contrast to traditional process planning and programming of manufacturing equipment at an early stage in the product development process, the required robot control code is here generated automatically and instantly (Adamson, Wang, and Moore 2017). This approach provides great flexibility since the functionality mapped into the AF-FB may be able to generate the same assembly result when performed by different assembly resources (Adamson et al. 2015). For this, different algorithms are created and included in the AF-FB, each customised to match specified robots, tools and assembly scenarios. A data input is then used to read e.g. the ID of the robot at AF-FB initialisation, for the selection of the corresponding algorithms.

Control approaches may range from reading robot target information from a single sensor, to be input to AF-FB algorithms generating robot move instructions for an *Insert* feature, to the complex processing of information from a network of sensors, actuators and controllers, by algorithm implemented AI technologies, in order to analyse and generate optimal robot path control for a robot operation (Adamson, Wang, and Moore 2017). It is possible for an AF-FB to generate legacy robot control instructions code to best utilise the legacy robot tools, before runtime models of the FBs become the built-in functions of robot controllers. AF-FBs could also be linked to external computational resources, as applied in Cloud Robotics, for support in calculating e.g. optimal robot paths.

Combining IEC 61499 FBs with MfgFs for realising adaptive equipment control solutions has been successfully demonstrated for some different manufacturing scenarios. In (Holm et al. 2012; Holm, Adamson, and Wang 2013; Wang, Hao, and Shen 2007; Wang, Holm, and Adamson 2010) Machining Features are used for CNC-machining control, and in (Adamson et al. 2012; Adamson, Holm, and Wang 2012; Wang et al. 2012, 2015) AFs are used for robotic assembly control.

5. Cloud manufacturing control structure

Since the introduction of Cloud Computing, with models to offer software, infrastructure, platforms and applications in the form of services, cloud technology has been extended to the manufacturing domain (Wu et al. 2015). Various distributed manufacturing systems have been presented (Tao et al. 2011; Xu 2012; Zhang et al. 2012), offering on-demand and scalable manufacturing services over the Internet from a shared pool of distributed manufacturing resources (Wu, Terpenny, and Schaefer 2016). These resources range from facilities, work-cells, machine tools and robots, to capabilities and software. The use of such manufacturing services within CM is fundamental for distributed manufacturing and also facilitates collaborative manufacturing missions (Bouzary and Chen 2018; Li et al. 2018; Moghaddam and Nof 2018).

The distributable nature of IEC 61499 Function Blocks is an important property for their use in distributed manufacturing environments, such as within CM. Besides different FB types, the IEC 61499 standard also defines the interaction and communication between distributed FBs. This enables networked MfgF-FBs to be integrated as manufacturing services in a cloud platform for the planning and execution of manufacturing tasks at different system control levels (Adamson et al. 2015). Integrating AF-FBs in such a control structure constitutes a cloud service for robot control, implementing the concept of Robot Control-as-a-Service (RCaaS). Supporting FBs of different types are needed in such control structures for control activities beyond the functionality of AF-FBs, to facilitate execution control and enable process monitoring during FB execution, as well as to enable communication between distributed FBs.

Amongst a multitude of possible cloud control scenarios for a manufacturing task request, two extreme alternative solutions exist (Adamson et al. 2015; Adamson, Wang, and Moore 2017):

- a complete high-level cloud service, for which one service provider performs all necessary manufacturing tasks, e.g. a provider offering Manufacturing/Assembly-as-a-Service in a resource-service Many-to-One mapping,
- a low-level cloud service approach, for which a combination of service providers each provide low-level services to collaboratively complete the high-level task. E.g. RCaaS together with Robot Software-as-a-Service (RSaaS) and Robot Hardware-as-a-Service (RHaaS), a combination of many One-to-One resource-service mappings. (RHaaS implies that a provider offers the use of a robot, which could be provided by dedicated manufacturing centres or by providers offering the sharing of spare equipment capacity).

Between these two extremes, a multitude of service composition solutions are also possible, depending on the division of the consumer request into separate services.

5.1. Robot control-as-a-service

Compared with conventional centralised process planning systems, this control approach can distribute decision-making. It builds on a two-level planning and control structure separating generic data from resource-specific, in which Supervisory Cloud Planning (SCP) performs generic process planning and Local Operation Planning (LOP) performs detailed shop-floor operation planning and execution. The high-level process plans are generic and portable to alternative robots, and only need to be generated once. The low-level operation plans are adaptive and optimal to the chosen available robot, and are generated at runtime to absorb the last-minute change on a dynamic shop-floor. The RCaaS control procedure is triggered by the reception of compiled assembly task information from the Cloud Service Management (CSM). This launches a sequence of internal activities, which performs the two-level FB-based planning procedure for the generation of an AF-FB based control structure: a Composite FB constituting the Assembly Process Plan (APP). In this process, generic and robot-specific information are separated into SCP and LOP, to enable efficient and smart decision-making (Adamson, Wang, and Moore 2017). Decision-making is supported by networked databases, whereas the latest monitoring information is made available to the LOP, for instantiating the FBs with real-time parameters and conditions, and performing execution control. This approach offers a high degree of adaptability to changes, as the planning is performed on demand, based on real-time information.

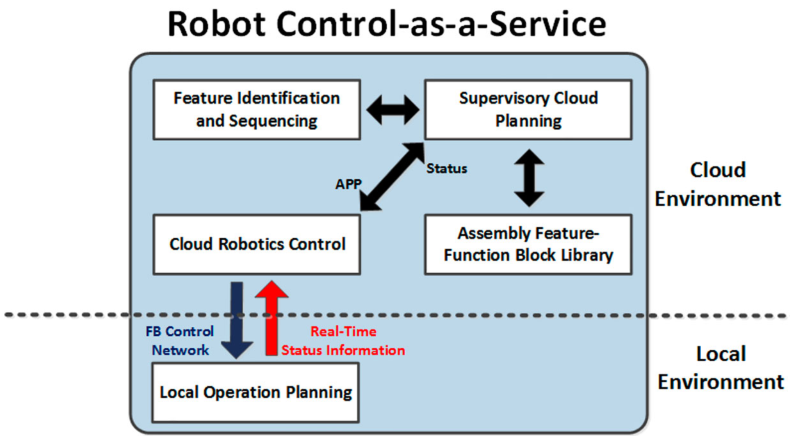

Figure 6. Robot Control-as-a-Service.

RCaaS consists of 5 cooperating modules, each performing different tasks (Adamson et al. 2015; Adamson, Wang, and Moore 2017), as seen in Figure 6.

- Supervisory Cloud Planning (SCP), (in the cloud)
- Feature Identification and Sequencing (FIS), (in the cloud)
- Assembly Feature-Function Block Library (AFL), (in the cloud)
- Cloud Robotics Control (CRC), (in the cloud)
- Local Operation Planning (LOP), (local, at the controlled resource(s)).

The control process includes the following steps and activities:

(1) SCP is performed once for the assembly task requested:

- AFs are identified and sequenced by the 'Feature Identification and Sequencing' module.
- By using pre-defined FBs from the AF-FB Library module, the SCP creates an APP by mapping necessary AF-FBs into a sequenced network of AF-FBs.

The APP created only contains necessary AF-FBs and their critical assembly sequence. This entails that it is generic and not tied to a specific robot. It can therefore be reused as well as ported to alternative robotic systems. (It is assumed that the activities of the Feature Identification and Sequencing module, AF recognition and sequencing, are performed and input to the SCP module. These activities are beyond the scope of this research).

(2) The Cloud Robotics Control module receives the APP from the SCP and has the following responsibilities:

- Construct the control structure,
- Distribute AF-FB control structures to the selected local shop-floors,
- Coordinate AF-FBs between different providers,
- Coordinate AF-FBs operation planning locally at each robot provider,
- Dynamic scheduling of resources and activities included,
- Perform robot initialisations,
- Perform FB execution control (start, stop, pause, resume, etc.),
- Monitor local robot execution, status and feedback to SCP,
- Update APP/AF-FBs in case of cloud level change (new/revised plans).

(3) Robot-level operation planning and execution, LOP:

- The generic APP is detailed through robot-level operation planning, as the embedded algorithms read their data inputs.
- Since LOP executes AF-FBs one by one, robot-specific control instructions are created at run-time through controller-level decision-making.

This control approach provides a high degree of adaptability to changes, by detailing the generic APP at LOP. Planning and execution are thus performed on demand, based on run-time information.

5.2. Cloud service management

The administration and supervisory management of all cloud services are performed by the CM platform CSM module (Figure 7). It is responsible for service discovery and matching, in which automatic decomposition of requested manufacturing tasks and composition of services to complete a task is one of the most attractive properties of CM. CSM is also responsible for dynamically coordinating manufacturing planning and execution control of distributed manufacturing resources. Dynamically coordinating services requires constant monitoring of run-time conditions and scheduled activities of all resources, which must all be accessible on-line.

To perform a manufacturing task, a variety of services in combinations are possible. In this research the focus is on the low-level cloud service approach alternative, to emphasise the multiple resource sharing and collaborative perspectives of CM, and it's inherent core virtue of creating higher levels of functionality, through the combination and composition of discrete services, which together can serve to complete consumers' high-level manufacturing task requests.

The CSM control procedure is initiated by a manufacturing task request from a resource consumer. The requests are analysed and divided into sub-tasks, and then distributed to matching manufacturing resources, for a coordinated manufacturing completion. This is supervised by the CSM, which selects and triggers the necessary services.

In the illustration of the CSM in Figure 7, three service providers participate to jointly deliver the required functionality to complete an assembly task request. It is assumed that a robot control provider supplies the robot control capability as RCaaS, and two providers supply the robot hardware as Robot Hardware-as-a-Service. The RCaaS is the instantiation of the MfgF-FBs control approach as a distributable service, and its executional control unit is a Composite FB. It acts as an APP including the necessary AF-FBs in the correct sequence to perform the assembly task.

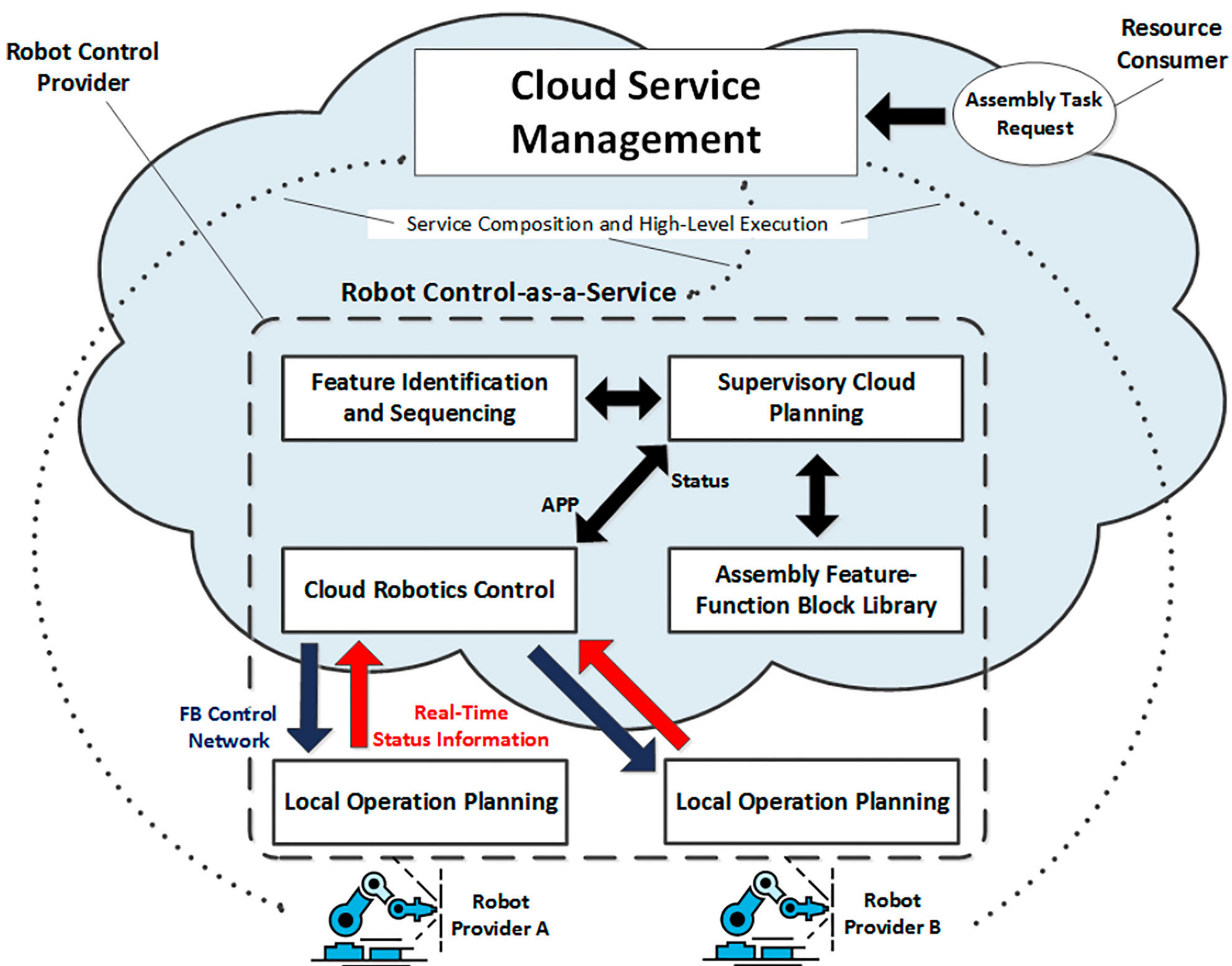

Figure 7. Robot Control-as-a-Service within CM environment.

6. Control concept demonstration

A control scenario for car engine assembly in a robotic assembly station is described, for which the FB control structure is presented. The assembly station may have different configurations regarding the number of robots, but the same control structure can be used to adaptively control different assembly stations, as it is defined to generate a certain generic functionality, i.e. the assembly of a car engine.

The assembly scenario presents the control of a virtual engine assembly station which in many regards mimics the behaviour of real engine assembly stations. Engines are gradually assembled in dedicated engine assembly lines, consisting of a number of unique assembly stations, connected by conveyors. Components to be assembled are placed on pallets which are transported through the complete assembly line by the conveyors. The majority of the assembly stations are fully automated, with robots performing different assembly tasks. Some stations are manually performed by operators, and some stations are semi-automated in which assembly operations are performed by both operators and robots. Engine block, engine head and pistons are mounted onto pallet fixtures, while other engine components to be assembled are fed directly to the assembly stations. Assembly is performed by mounting components onto/into a base unit, i.e. the engine block, and solely executed in the z-plane, meaning components will be placed/inserted/screwed onto/into the engine block in a top-down approach. The locations of components to be assembled are pre-defined within stations through the use of feeders and pallet. This information is available through the local Assembly Station DB, as the component pick locations, to be fed to the AF-FBs. (In most cases, these components could also be station sensor-detected.) The engine block's Base Frame location in station is therefore known, and signal triggered when pallet is in assembly position. AFs' locations are defined relative to the base unit's Base Frame, in the AF description in the consumer Product Data Model. This information details the locations to place, insert or screw components and is also fed to the AF-FBs. Pallets enter the assembly station on the conveyor one at a time, and when the station assembly operations have been performed, the pallet is forwarded to the next station, as a new pallet enters the station.

6.1. Engine assembly components

The following components are included in the engine assembly task (Figure 8):

Engine block: Fed to the station on pallet. AFs' locations are defined in the customer's Product Data Model, in relation to the engine block's Base Frame.

Engine head: Fed to the station on pallet.

Pistons: Complete with pin and rod. Fed to the station on pallet, pre-mounted on piston fixture.

Bolts: Bolts are separately fed to the station, pre-mounted in a bolt-fixture.

6.2. Engine assembly task

An assembly task specifies the operations needed to assemble a product, and a robotic assembly task can be performed by one or more robots in one or a group of cooperating robot stations/cells. In its simplest form an assembly task may be to assemble only two parts to create a sub-assembly, or as in this scenario, to assemble the above set of components into a car engine:

This high-level assembly task is realised by using a group of pre-defined AF-FBs, sequenced in a Composite FB, acting as an engine APP. As each AF-FB contains the detailed know-how of how to perform a low-level assembly operation, higher-level assembly tasks as engine assembly can be performed. In the case of a multi-robot station, cooperatively performing an engine assembly task implies that individual robots can perform individual AF-FBs of an APP. However, robots cannot share the execution of the same AF-FB, i.e. if one robot breaks down during the execution of an AF-FB, another robot cannot complete its operations. To ensure safety in the case two or more robots are cooperating to complete an assembly

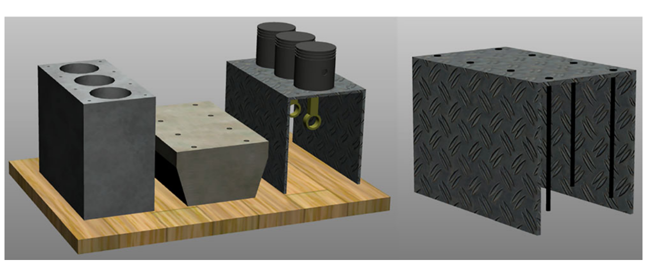

Figure 8. Engine assembly components.

task, a control structure functionality is required to make sure that one robot has finished its operations before the next robot can start. For this a safe Home location can be used for each robot in combination with the triggering of an output event to the next waiting robot, acknowledging that the first robot has reached it's safe location.

The AF-FBs required for this engine assembly task are:

- INSERT: to insert pistons into the engine block,
- PLACE: to place engine head onto engine block, and
- SCREW: to attach engine head to engine block with bolts.

6.3. Engine assembly task control

During SCP, before Assembly task execution can be initiated, an extended FB network is formed, i.e. the FB Control Structure. An APP is input to this structure, in relation to selected assembly task. To deal with those issues that cannot be handled by individual AF-FBs, such as interfacing with device controllers and control panels (HMIs), coordinating inter-FB activities and communication between distributed FBs, the following Service Interface-FBs are also added to the control structure:

- Cloud Control-Function Block (CC-FB): Acts as a manager on the cloud level.
- Local Control-Function Block (LC-FB): Acts as a manager on the local level.
- Material Handling-Function Block (MH-FB): Controls the material handling in the Assembly Station.
- Communication-Function Block (C-FB): Acts as an interface for communication between FBs at different levels.

These FBs are all of the Service Interface type which is defined in the IEC 61499 standard. The control structure also incorporates some assembly task information sources: Consumer and Assembly Station DBs and the CSM module.

6.4. Function block control structure

In Figure 9 the main details of the FB control structure and its connections are shown, together with its information sources.

- Cloud Control-Function Block
  Responsible for managing assembly task cloud level activities, such as: interface the CSM module, coordinate activities between participating service providers, scheduling of resources, perform robot initialisations and FB execution control (start, stop, pause, resume, etc.), perform assembly task simulation, monitor local task execution, and in the case of changed task requirements (e.g. change in product design), update APP.
- Local Control-Function Block
  Responsible for managing assembly task activities at the local Assembly Station level, such as: LOP and FB execution, interfacing with CC-FB, station device controllers and HMIs, coordination and monitoring of activities to accomplish the assembly task, conducting local equipment initialisation, transmitting AF-FBs generated control commands to the robot controller, and passing updates to FBs.
  The LC-FB is downloaded to a front-end station controller, with an IEC 61499 run-time environment for execution of FBs. This controller interfaces the Assembly Station through a local network (LAN) for access to robot controller and station equipment, e.g. conveyor and station sensors. As shown in Figure 9, the LC-FB is connected to the AF-FBs (APP) and the MH-FB, while interfacing and sharing runtime data with the assembly station. This facilitates runtime information retrieval from the assembly station and communicating this information to the control generating AF-FBs.
- Assemble Engine-Function Block (AE-FB)
  In this scenario, the APP contains three sequenced AF-FBs to control the robotic task of assembling a car engine. At run time execution, each AF-FB generates detailed operation plans as their algorithms generate the required robot control instructions. These instructions are forwarded to the robot controller(s) through the coordinating LC-FB. After the execution of one AF-FB, the next is called according to the sequence defined in the APP. During assembly operation execution, process monitoring is crucial for successfully completing the assembly task. Based on real-time monitoring data, it is also possible to effectively coordinate resource selection, job dispatching and process execution. Therefore, all of the FBs, upon request to enable monitoring, can convey status information to the CC-FB.
- Material Handling-Function Block

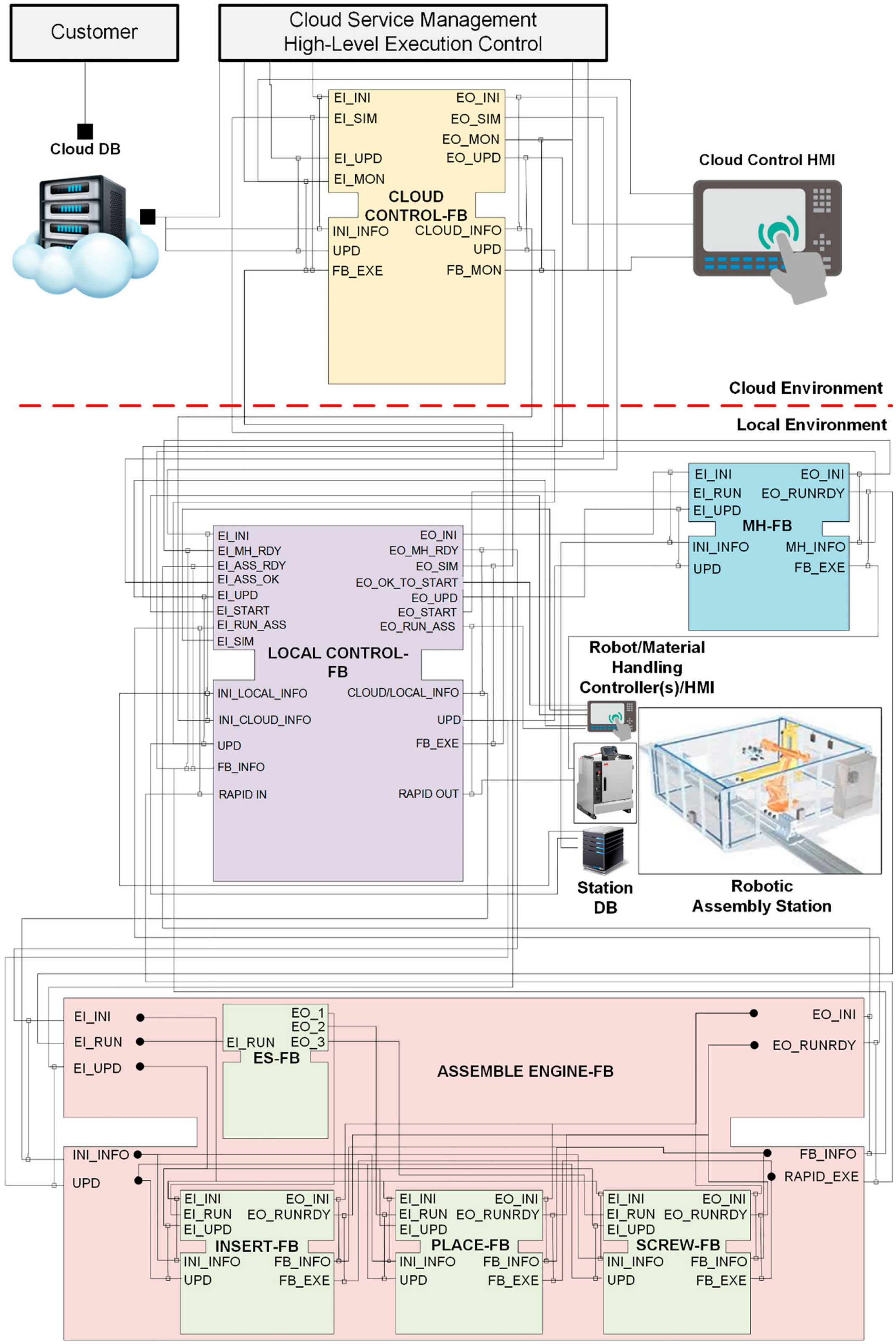

Figure 9. Function Block control structure.

Another type of Service Interface-FB, called Material Handling-Function Block, (MH-FB) is used as a controller for the conveyor system, controlling the flow and monitoring the locations of pallets available in the assembly station.

- Communication-Function Block
  A Communication-Function Block is a Service Interface FB that provides a construct for information sharing between distributed FBs. It is designed to facilitate FB execution and process monitoring by providing necessary communications between computers for planning and control and controllers on shop floors. It is extended from the Service Interface FB type defined in IEC-61499, and can provide services including FB dispatching and runtime operation status sharing through a distributed network. Communication-FBs are not shown in the control structure (Figure 9), but are situated between cloud and local levels.
- Cloud Service Management (CSM)
  Responsible for service management, e.g. service composition and high-level control, as part of the CM concept. Coordinating the execution for selected service providers for a manufacturing task.
- Customer Cloud DB
  Provides the consumer feature-enriched Product Data Model expressing the required MfgFs, process parameters and requirements for product creation.
- Robotic Assembly Station DB
  Provides status information for the Robotic Assembly Station such as robot and tool IDs, component locations, conveyor and station run-time status, etc.

6.5. *Control structure functionality*

Before assembly can start, the control structure needs to be initialised with the actual assembly task and Assembly Station information. The core of the assembly process control is the LC-FB. At the very beginning, it is being asked to initialise by receiving an output event from the CSM, EO_INI. This triggers the initialisation of the LC-FB, as it reads the local Assembly Station information from the Station DB, and the cloud product information from the Consumer Cloud DB. When initialisation is finished, an output initialisation event is sent to the MH-FB, which is also initialised by the Station DB. After this, the AE-FB is similarly initialised, as it is triggered by EO_MH_RDY to read the compiled cloud and local information from the LC-FB. As the EI_ASS_RDY is received from the AE-FB, the control structure initialisation process is concluded, which is now prepared to commence the engine assembly task, either as simulation mode or actual assembly operations (EI_SIM or EI_START activated by operator HMI). Simulation is performed to verify a correct and collision-free robot path before the real run is started (Adamson et al. 2014). Other parameters of interest for high-level planning and scheduling, such as cycle time, can also be verified. In this scenario, the Assembly Station provider has uploaded a cloud level simulation model, as part of the service offered, but the simulation could also be performed on the local level.

The actual assembly task is started as EO_START triggers the MH-FB to forward a loaded pallet to the assembly position. After this, the robot control instructions for the assembly operations are generated by the three AF-FBs in the AE-FB, which is triggered by EI_RUN. Each AF-FB generates detailed operation plans at run time, as the it's algorithms execute. The LC-FB receives these control instructions and is in charge of conveying these to a robot controller. Depending on the availability of robots, it can decide which one should be selected.

The FBs on all levels can be updated when assembly conditions change. This information is conveyed by the EI_UPD event input coupled with the UPD data input. The sources for these changes can be both product and station related. Cloud level monitoring of the ongoing process is also possible, and is initiated by the EI_MON input event to the CC-FB.

6.6. *Assembly task sequence*

At Assembly Station start-up, local control FBs are initialised, reading actual conditions regarding assembly task and assembly station. Information such as AF locations, robot operational parameters, e.g. target locations, speed, safety levels, are then read by the AF-FB's data inputs. This information is then used by the algorithms to create a run-time adapted control. A change of the assembly task, e.g. a change in product design which results in a new location for an AF, can be directly conveyed to the actual AF-FB through the update event functionality. This information is then accessed directly from the customer's Cloud DB.

In the following description off the sequenced execution of the engine assembly task, references are given to the FBs in the control structure generating the control, as well as details of separate task operations. (For the first AF procedure, Inserting Pistons, figures describing the assembly task information flow is presented. The information flow for the other AFs

Engine Assembly Task

- Feature Sequence:
  - INSERT
    * **Open tool** – Release, Vacuum Off
    * **Move I1** – Above Piston Fixture
    * **Move I2** – Down to pick location, Piston Fixture
    * **Close tool** – Pick Pistons, Vacuum On
    * **Move I3** - Lift Pistons vertically (to safe position)
    * **Move I4** - Above Engine Block
    * **Move I5** – Approach Engine Block (with angle)
    * **Move I6** – Down to insertion Location
    * **Move I7** – Insert Pistons (Insertion Depth value)
    * **Open tool** – Release Pistons, Vacuum Off
    * **Move I8** – Above Engine Block
    * **Move I9** – Home
  + PLACE
  + SCREW

Figure 10. INSERT Assembly Feature operations.

(Place Engine Head, and Screw Bolts) is performed in the same manner). For ease of understanding in this presentation, all parameters are not included in the feature descriptions:

(1) A pallet enters the station, and is detected by station sensors: first at station entry, and next when it has reached the pre-defined assembly location. Controlled by the MH-FB.
(2) The pistons are inserted into the engine block. The robot control instructions are generated by the INSERT AF-FB (Figure 10):

Piston pick location I2, given by Station DB (Figure 11). Robot target location I2 initialised or updated by Station DB.
Safety level given by Station DB (optional) (Figure 12).

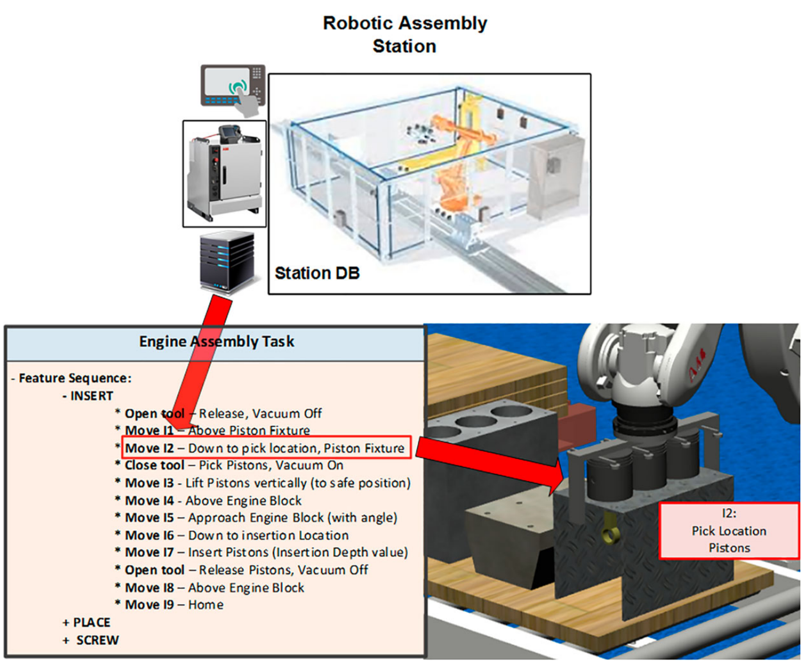

Figure 11. Robot target location I2 received from Station DB.

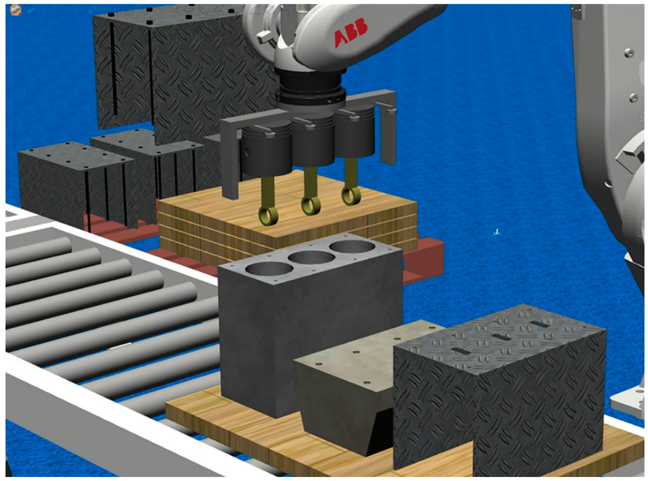

Figure 12. Piston safe location.

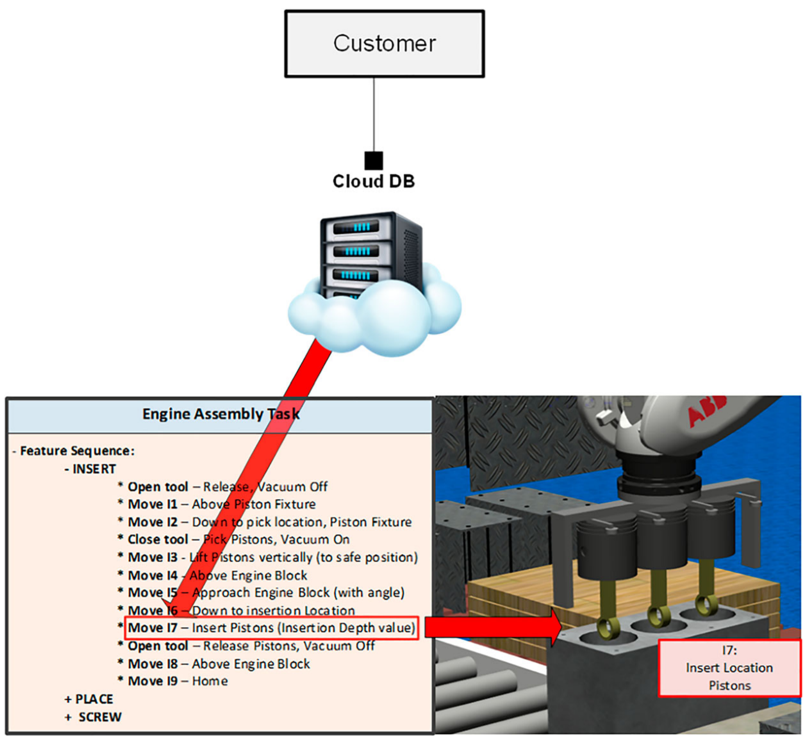

Figure 13. Target location I7 received from Customer Cloud DB.

Insert feature location and depth given by consumer's Cloud DB, at initialisation or update (Figure 13). Location in relation to engine block Base Frame (defined in Station DB).

(3) The engine head is placed onto the engine block. The robot control instructions are generated by the PLACE AF-FB:

Engine Head pick location given by Station DB. Safety level given by Station DB (optional). Place location given by consumer's Cloud DB. Location in relation to engine blocks Base Frame (defined in Station DB) (Figure 14).

(4) The bolts are screwed through the engine head, down into the engine block. The robot control instructions are generated by the SCREW AF-FB:

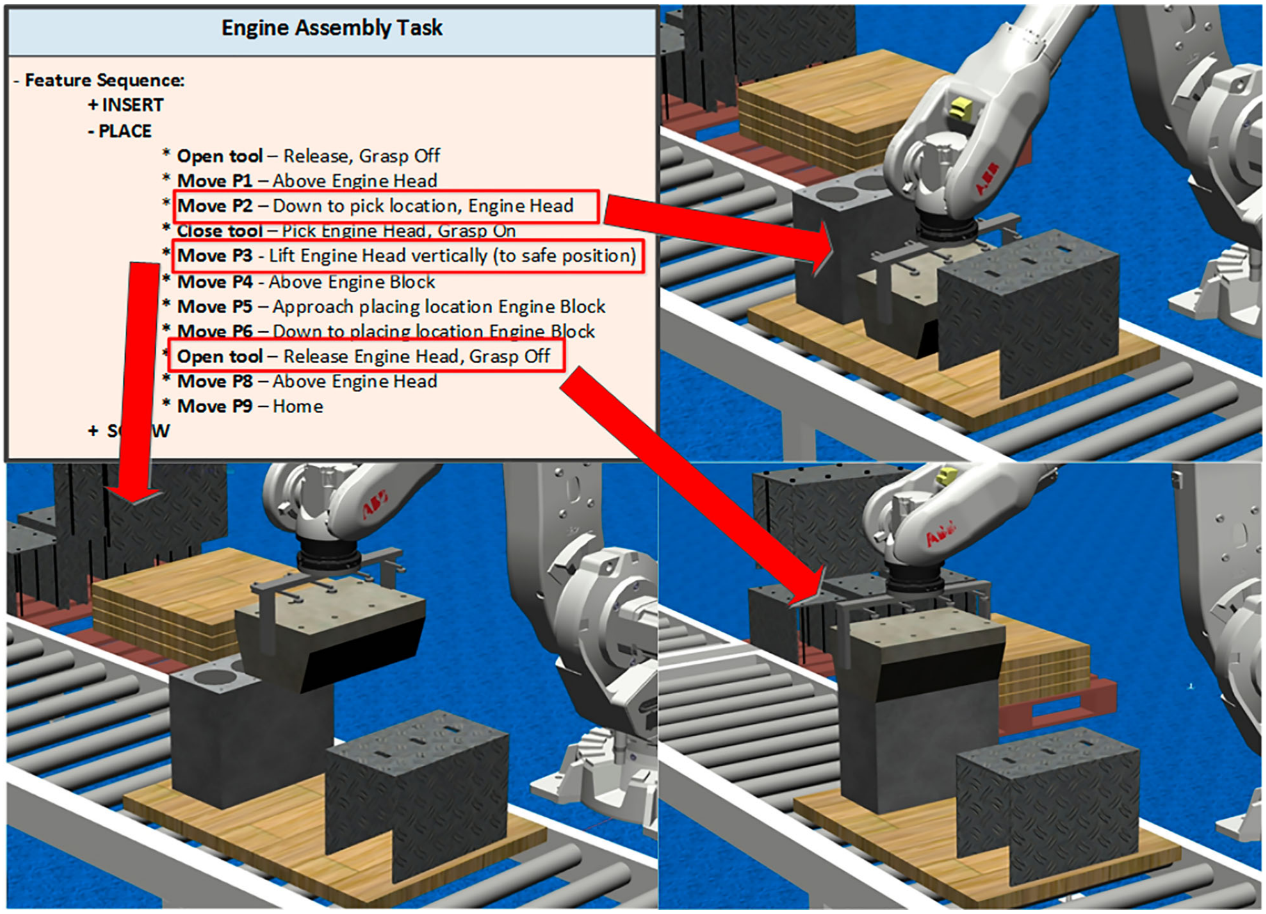

Figure 14. Operations for placing engine head onto engine block.

Bolt pick location given by station DB, safety level given by Station DB (optional), Bolt mounting location given by consumer's Cloud DB, in relation to engine blocks Base Frame. Additional parameters for screw process also included, e.g. bolt torque (Figure 15).

(5) Finished engine leaves station as new pallet enters. Controlled by the MH-FB.

6.7. Adaptivity scenarios – handling variations

The sources for possible variations in performing the engine assembly are mainly related to either the engine (Product) or the Assembly Station (Resource):

- Product design change (e.g. a component feature location has been moved):
  If the new location is within the capability of the selected robot, an update of the location from the Product Model in the Customer Cloud DB, or through the CC-FB, is enough. (If LC-FB continuously reads component feature locations from CC-FB or Customer DB, changed locations will be handled automatically. If LC-FB reads all component feature locations at initialisations, as in the described scenario, update of changed location is necessary.)
  If not within the available capability, a new resource needs to be selected by CSM.
  If there is a change of the number of components, SCP needs to be performed, for selecting and sequencing the correct set of AF-FBs.
- Product process change (e.g. bolt torque has been increased):
  If the new assembly process parameters is within the capability of the selected robot station, an update of the process parameter from the Customer Cloud DB or through the CC-FB is enough.
  If not within the available capability, a new resource needs to be selected by CSM.

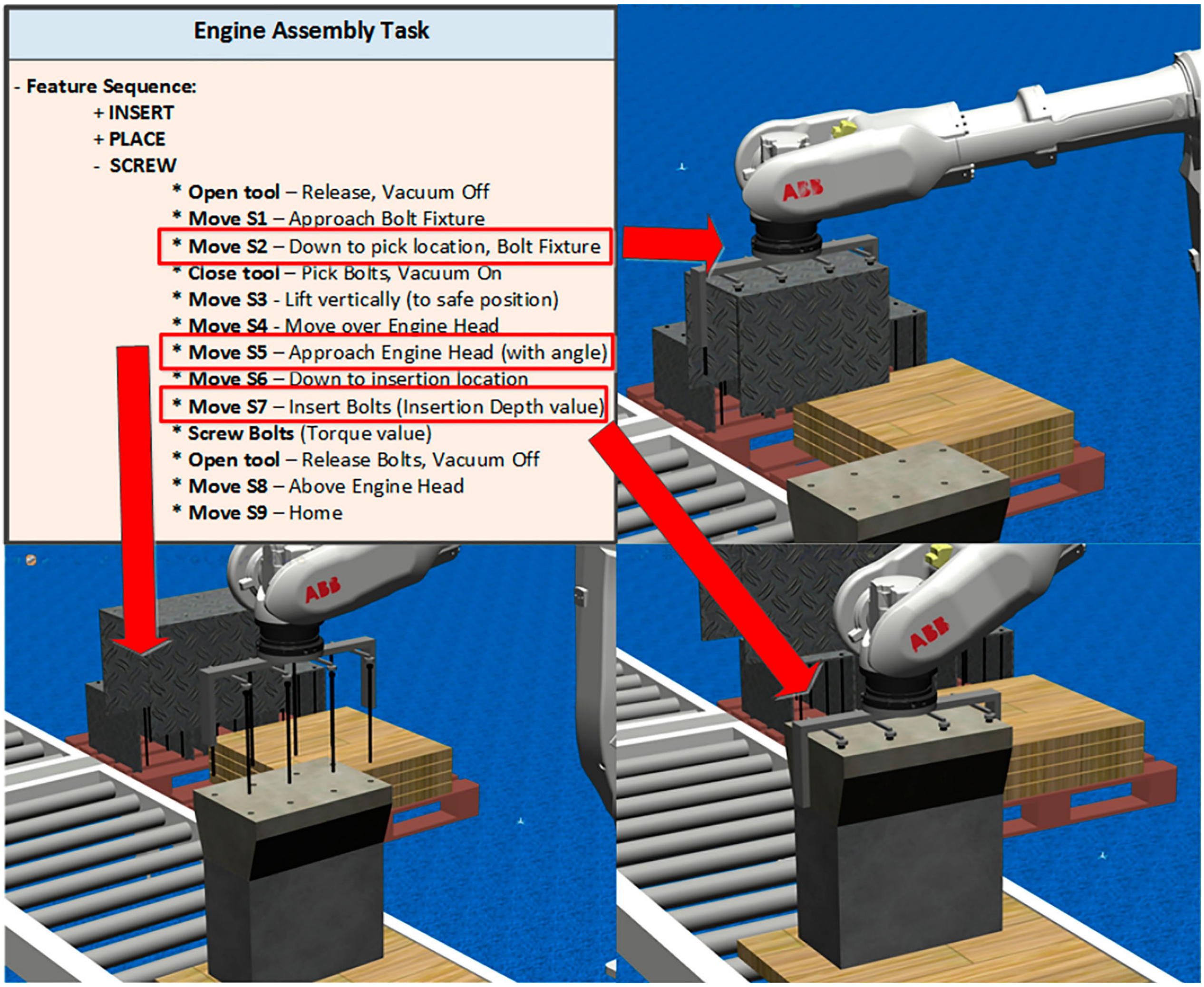

Figure 15. Operations to screw bolts through engine head into engine block.

- Manufacturing resource variations:

  The FB control structure can also handle some variations in station equipment. If there is a resource redundancy within the manufacturing system, sometimes another resource can be automatically invoked as a replacement for a failing resource.

  In the case where two robots are cooperating to complete an assembly task and one of them becomes unavailable (service, breakdown, etc.), the control structure can re-direct all control commands to the available robot. The Assembly Station DB holds the record 'Active/Not active' for each resource, so the LC-FB can select which robot ID to convey the control instructions to. This is possible if the 'replacing' robot's capability matches the capability of the unavailable robot, for the actual operational parameters (reach, weight, tool, etc.).

  To improve this ability, an optional Tool Change-Function Block can also be included in the control structure. This can be triggered in case of an unavailable or broken tool, and should include the necessary robot movements and signals for a tool change.

7. Conclusions and future work

Implementations of the proposed control approach prove that it has many promising characteristics for use within both local and distributed environments, such as cloud environments. The biggest advantage compared to traditional control is that the required control is created at run-time according to actual manufacturing conditions, facilitating rapid adaptation to the changes in product design, assembly conditions and assembly environment.

The biggest obstacle for being applicable to its full extent is manufacturing equipment controlled by proprietary control systems, with native control languages. Such controllers might provide different levels of access to its internal data and commands, restricting external control functionality. To take the full advantage of the IEC FB control approach, controllers which can interface, interpret and execute these FBs directly, are necessary.

A structured library with Feature-Function Blocks for different and complex manufacturing tasks and operations also needs to be developed, to facilitate the automated generation of adequate Feature-Function Block process plans. The effective mapping of different atomic manufacturing operations into unique Manufacturing Features, therefore, needs to be established.

Automated assembly process planning, for the automatic generation of an APP for a selected assembly task, is also necessary. The correct AF-FB then need to be selected and sequenced. (Process planning for other manufacturing operations are performed similarly as for assembly operations). These plans should be able to handle dynamic assembly environments since the execution of an APP and its AF-FB is based on real-time manufacturing information and is, therefore, able to handle variations in an adaptive manner.

Since the generated output data from the Function Block algorithms is used to control the robotic operations, Function Block algorithm development is of major concern. For optimal robot path generation, the algorithms could be constructed to link to external cloud services offering methods for robot path calculations, such as Simulation-based Optimisation. It would then be possible to find the best solutions for specific task requirements e.g. cycle time, energy consumption, interfacing humans, shortest path, etc. This approach could also be used for the optimal sequencing of assembly operations for complex assembly tasks.

The supporting Feature-based Information Framework presented enables the matching of manufacturing task requests with provider resources' capabilities and as such only addresses the issue of functional capabilities for manufacturing resources. Many other aspects of capability are of course also of interest in the matching of manufacturing tasks and resources. Resource properties such as cost, quality, capacity, availability, delivery times, resource location, customer ratings, etc., may also be expressed in manufacturing capability models. The functional capability answers the question of 'what' the resource can perform in regard to production capacity, while other capability descriptions answers the question of 'how', in respect to resource properties. When a manufacturing task request is published within a CM platform, it ought to be possible for the Consumer to estimate and compare different proposed service solution on the basis of more criteria than only the functional capabilities of manufacturing resources. Therefore, extended manufacturing resource capability models need to be established, enabling retrieval of the best service solution in relation to a group of desired resource properties. The use of simulation techniques to solve such a multi-objective optimisation problem could be one possible approach.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

The authors are thankful to the Swedish Knowledge Foundation [grant number 20160297] for providing support to accomplish the presented work.

ORCID

Göran Adamson http://orcid.org/0000-0003-1265-8451
Lihui Wang http://orcid.org/0000-0001-8679-8049

References

Adamson, Göran, Magnus Holm, and Lihui Wang. 2012. "Event-Driven Adaptability Using IEC 61499 in Manufacturing Systems." In *The 5th International Swedish Production Symposium, SPS12, 6–8 November 2012, Linköping, Sweden*, edited by Björkman Mats, 453–460. Linköping: The Swedish Production Academy.

Adamson, Göran, Magnus Holm, Lihui Wang, and Philip Moore. 2012. "Adaptive Assembly Feature Based Function Block Control of Robotic Assembly Operations." In *Paper Presented at the 13th Mechatronics Forum International Conference*. Linz: Johannes Kepler University, Monday, September 17, 2012 to Wednesday, September 19, 2012.

Adamson, Göran, Lihui Wang, Magnus Holm, and Philip Moore. 2014. "*Function Block Approach for Adaptive Robotic Control in Virtual and Real Environments*." Paper presented at the proceedings of the 14th mechatronics forum international conference.

Adamson, Göran, Lihui Wang, Magnus Holm, and Philip Moore. 2015. "Adaptive Robot Control as a Service in Cloud Manufacturing." (56833): V002T04A20. doi:10.1115/MSEC2015-9479.

Adamson, Göran, Lihui Wang, and Philip Moore. 2017. "Feature-based Control and Information Framework for Adaptive and Distributed Manufacturing in Cyber Physical Systems." *Journal of Manufacturing Systems* 43 (Part 2): 305–315. doi:10.1016/j.jmsy.2016.12.003.

Boutellier, Roman, Oliver Gassmann, and Maximilian von Zedtwitz. 2008. *Managing Global Innovation, Uncovering the Secrets of Future Competitiveness*. 3rd ed. Berlin: Springer.

Bouzary, Hamed, and F. Frank Chen. 2018. "Service Optimal Selection and Composition in Cloud Manufacturing: a Comprehensive Survey." *The International Journal of Advanced Manufacturing Technology* 97: 1–14.

Brennan, R. W., X. Zhang, Y. Xu, and H. Norrie. 2002. "A Reconfigurable Concurrent Function Block Model and its Implementation in Real-Time Java." *Integrated Computer-Aided Engineering* 9 (3): 263–279.

Ding, Bo, Xiao-Yang Yu, and Li-Juan Sun. 2012. "A Cloud-Based Collaborative Manufacturing Resource Sharing Services." *Information Technology Journal* 11 (9): 1258.

Doukas, George S, Kleanthis C Thramboulidis, and Yannis C Koveos. 2006. "*Using the Function Block Model for Robotic Arm Motion Control*." Paper presented at the control and automation, 2006. 14th Mediterranean conference on MED'06.

Groover, Mikell P. 2016. Automation, Production Systems, and Computer-integrated Man|ufacturing: Pearson Education India.

Herterich, Matthias M., Falk Uebernickel, and Walter Brenner. 2015. "The Impact of Cyber-Physical Systems on Industrial Services in Manufacturing." *Procedia CIRP* 30: 323–328. doi:10.1016/j.procir.2015.02.110.

Holm, Magnus, Göran Adamson, and Lihui Wang. 2013. "Enhancing Adaptive Production Using IEC 61499 Event-Driven Function Blocks." In *41st North American Manufacturing Research Conference 2013, NAMRC 2013,* 10 June 2013 through 14 June 2013, 420–429. Madison, WI: Society of Manufacturing Engineers, North American Manufacturing Research Institute of SME.

Holm, Magnus, Göran Adamson, Lihui Wang, and Philip Moore. 2012. "An IEC 61499 Function Block Based Approach for CNC Machining Operations." In *13th Mechatronics Forum International Conference, Johannes Kepler University, Linz, Monday, September 17, 2012 to Wednesday, September 19, 2012*, edited by Scheidl Rudolf, and Jakoby Bernhard, 115–121. Linz: TRAUNER Verlag.

Huang, Xuemei. 2010. "*Enhancing STEP-NC Compliant CNC Controller for Distributed and Reconfigurable Environment in Production Line*." Paper presented at the 2010 international conference on computer, mechatronics, control and electronic engineering (CMCE).

IEC, IEC. 2005. "61499-1, Function Blocks-Part 1: Architecture." *International Standard, First Edition, International Electrotechnical Commission, Geneva* 1:2005.

ISO. 2007. "ISO 10303-238 STEP for numerical control."

Kusiak, Andrew. 2017. "Smart Manufacturing." *International Journal of Production Research*, 1–10. doi:10.1080/00207543.2017.1351644.

Lewis, Robert. 2001. *Modelling Distributed Control Systems Using IEC 61499: Applying Function Blocks to Distributed Systems*. London: IEE.

Li, Changyi, Jianhe Guan, Tingting Liu, Ning Ma, and Jun Zhang. 2018. "An Autonomy-Oriented Method for Service Composition and Optimal Selection in Cloud Manufacturing." *The International Journal of Advanced Manufacturing Technology* 96: 1–22.

Li, B. H., L. Zhang, and X. Chai. 2010. "Introduction to Cloud Manufacturing." Accessed 4 December. http://wwwen.zte.com.cn/endata/magazine/ztecommunications/2010Year/no4/articles/201012/t20101219_196752.html

Meier, Matthias, Joachim Seidelmann, and István Mezgár. 2010. "ManuCloud: the Next-Generation Manufacturing as a Service Environment." *ERCIM News* 83: 33–34.

Minhat, Mohamad, Valeriy Vyatkin, Xun Xu, Soon Wong, and Zina Al-Bayaa. 2009. "A Novel Open CNC Architecture Based on STEP-NC Data Model and IEC 61499 Function Blocks." *Robotics and Computer-Integrated Manufacturing* 25 (3): 560–569.

Minhat, Mohamad, Xun Xu, and Valeriy Vyatkin. 2009. "STEPNCMillUoA: a CNC System Based on STEP-NC and Function Block Architecture." *International Journal of Mechatronics and Manufacturing Systems* 2 (1–2): 3–19.

Moghaddam, Mohsen, and Shimon Y Nof. 2018. "Collaborative Service-Component Integration in Cloud Manufacturing." *International Journal of Production Research* 56 (1–2): 677–691.

Monostori, László, B. Cs Csáji, Botond Kádár, András Pfeiffer, E. Ilie-Zudor, Zs Kemény, and Marcell Szathmári. 2010. "Towards Adaptive and Digital Manufacturing." *Annual Reviews in Control* 34 (1): 118–128.

Olsen, Scott, James Wang, Alejandro Ramirez-Serrano, and Robert W. Brennan. 2005. "Contingencies-based Reconfiguration of Distributed Factory Automation." *Robotics and Computer-Integrated Manufacturing* 21 (4-5): 379–390.

Selvaraj, P., P. Radhakrishnan, and M. Adithan. 2009. "An Integrated Approach to Design for Manufacturing and Assembly Based on Reduction of Product Development Time and Cost." *The International Journal of Advanced Manufacturing Technology* 42 (1): 13–29. doi:10.1007/s00170-008-1580-8.

Strasser, Thomas, Franz Auinger, and Alois Zoitl. 2004. "*Development, Implementation and use of an IEC 61499 Function Block Library for Embedded Closed Loop Control*." *Paper presented at the industrial informatics, 2004. 2nd IEEE international conference on INDIN'04. 2004*.

Strasser, Thomas, Alois Zoitl, James H. Christensen, and Christoph Sünder. 2011. "Design and Execution Issues in IEC 61499 Distributed Automation and Control Systems." *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 41 (1): 41–51.

Tao, F., L. Zhang, V. C. Venkatesh, Y. Luo, and Y. Cheng. 2011. "Cloud Manufacturing: A Computing and Service-Oriented Manufacturing Model." *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* 225 (10): 1969–1976. doi:10.1177/0954405411405575.

Thramboulidis, Kleanthis, and Chris Tranoris. 2001. "*An Architecture for the Development of Function Block Oriented Engineering Support Systems*." *Paper presented at the computational intelligence in robotics and automation, 2001. IEEE international symposium on proceedings 2001*.

Valilai, Omid Fatahi, and Mahmoud Houshmand. 2013. "A Collaborative and Integrated Platform to Support Distributed Manufacturing System Using a Service-Oriented Approach Based on Cloud Computing Paradigm." *Robotics and Computer-Integrated Manufacturing* 29 (1): 110–127. doi:10.1016/j.rcim.2012.07.009.

Völker, Norbert, and Bernd J Krämer. 2002. "Automated Verification of Function Block-Based Industrial Control Systems." *Science of Computer Programming* 42 (1): 101–113.

Vyatkin, Valeriy. 2011. "IEC 61499 as Enabler of Distributed and Intelligent Automation: State-of-the-art Review." *IEEE Transactions on Industrial Informatics* 7 (4): 768–781.

Wang, Lihui, Göran Adamson, Magnus Holm, and Philip Moore. 2012. "A Review of Function Blocks for Process Planning and Control of Manufacturing Equipment." *Journal of Manufacturing Systems* 31 (3): 269–279. doi:10.1016/j.jmsy.2012.02.004.

Wang, Lihui, Robert W. Brennan, Sivaram Balasubramanian, and Douglas H Norrie. 2001. "Realizing Holonic Control with Function Blocks." *Integrated Computer-Aided Engineering* 8 (1): 81–93.

Wang, Lihui, Qi Hao, and Weiming Shen. 2007. "A Novel Function Block Based Integration Approach to Process Planning and Scheduling with Execution Control." *International Journal of Manufacturing Technology and Management* 11 (2): 228–250. doi:10.1504/ijmtm.2007.013193.

Wang, L., M. Holm, and G. Adamson. 2010. "Embedding a Process Plan in Function Blocks for Adaptive Machining." *CIRP Annals* 59 (1): 433–436. doi:10.1016/j.cirp.2010.03.144.

Wang, Lihui, Bernard Schmidt, Mohammad Givehchi, and Göran Adamson. 2015. "Robotic Assembly Planning and Control with Enhanced Adaptability Through Function Blocks." *The International Journal of Advanced Manufacturing Technology* 77 (1–4): 705–715.

Wang, Lihui, and Xi Vincent Wang. 2018. "Adaptive Machining Using Function Blocks." In *Cloud-Based Cyber-Physical Systems in Manufacturing*, 125–162. Cham: Springer International Publishing.

Wang, Hongqiang, Xun Xu, and J. Des Tedford. 2007. "An Adaptable CNC System Based on STEP-NC and Function Blocks." *International Journal of Production Research* 45 (17): 3809–3829.

Wu, Dazhong, David W Rosen, Lihui Wang, and Dirk Schaefer. 2015. "Cloud-based Design and Manufacturing: A New Paradigm in Digital Manufacturing and Design Innovation." *Computer-Aided Design* 59: 1–14.

Wu, Dazhong, Janis Terpenny, and Dirk Schaefer. 2016. "Digital Design and Manufacturing on the Cloud: A Review of Software and Services." *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, *Cambridge University Press* 31: 1–15.

Wu, Lei, and Chengwei Yang. 2010. "A Solution of Manufacturing Resources Sharing in Cloud Computing Environment." In *Cooperative Design, Visualization, and Engineering*, 247–252. Berlin, Heidelberg: Springer.

Xu, Xun. 2012. "From Cloud Computing to Cloud Manufacturing." *Robotics and Computer-Integrated Manufacturing* 28 (1): 75–86. doi:10.1016/j.rcim.2011.07.002.

Xu, Xun, Lihui Wang, and Stephen T Newman. 2011. "Computer-aided Process Planning–A Critical Review of Recent Developments and Future Trends." *International Journal of Computer Integrated Manufacturing* 24 (1): 1–31.

Yao, Bitao, Zude Zhou, Lihui Wang, Wenjun Xu, Junwei Yan, and Quan Liu. 2018. "A Function Block Based Cyber-Physical Production System for Physical Human–Robot Interaction." *Journal of Manufacturing Systems*. doi:10.1016/j.jmsy.2018.04.010.

Zhang, Lin, Yongliang Luo, Fei Tao, Bo Hu Li, Lei Ren, Xuesong Zhang, Hua Guo, Ying Cheng, Anrui Hu, and Yongkui Liu. 2012. "Cloud Manufacturing: a new Manufacturing Paradigm." *Enterprise Information Systems* 8 (2): 167–187. doi:10.1080/17517575.2012.683812.

Zoitl, Alois. 2008. Real-Time Execution for IEC 61499: ISA.

Zoitl, Alois, Gunnar Grabmair, Franz Auinger, and C. Sunder. 2005. "*Executing Real-Time Constrained Control Applications Modelled in IEC 61499 with Respect to Dynamic Reconfiguration*." Paper presented at the industrial informatics, 2005. 3rd IEEE international conference on INDIN'05, 2005.