ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/267391638

OPC UA for machine tending industrial robots – Prototypic development of an OPC UA server for ABB industrial robots

Conference Paper · October 2014

DOI: 10.15224/978-1-63248-031-6-155

CITATIONS
8

READS
4,390

3 authors:

Florian Pauker
University of Vieanna
25 PUBLICATIONS **196** CITATIONS
SEE PROFILE

Iman Ayatollahi
TU Wien
14 PUBLICATIONS **97** CITATIONS
SEE PROFILE

Burkhard Kittl
TU Wien
22 PUBLICATIONS **162** CITATIONS
SEE PROFILE

Some of the authors of this publication are also working on these related projects:

model-UA View project

centurio.work View project

All content following this page was uploaded by Florian Pauker on 29 January 2015.

The user has requested enhancement of the downloaded file.

Proc. of the Second Intl. Conf. on Advances In Mechanical and Robotics Engineering- AMRE 2014.
Copyright © Institute of Research Engineers and Doctors, USA .All rights reserved.
ISBN: 978-1-63248-031-6 doi: 10.15224/ 978-1-63248-031-6-155

OPC UA for machine tending industrial robots

Prototypic development of an OPC UA server for ABB industrial robots

[Florian Pauker, Iman Ayatollahi, Burkhard Kittl]

***Abstract* — In this paper an information model and a prototype OPC UA server using this model for data acquisition, event generation and remote control of a machine tending industrial robot is described. Necessary information is transferred from the ABB machine tending software, mapped to appropriate nodes of the information model, and exposed by the OPC UA server. The authors believe that OPC UA with its extensible modelling potential is well suited as a communication technology in this field of manufacturing automation. OPC UA is still not being applied to its full potential in this area. Occasionally some industry driven efforts are observed to map existing data models to the most recent OPC specifications (UA), but the development of semantic rich information models is mainly neglected.**

For demonstration purpose a prototypic OPC UA server with a simple information model for an ABB IRB 120 industrial robot was realized. The robot controller provides the server with data by using the ABB PC Interface. This interface enables a server to operate with one or more (either simulated virtual or real) robot controllers.

***Keywords* — OPC UA, robot, machine tending, ABB, communication interface**

I. Introduction

The process of loading and unloading machine tools with work pieces is referred to as machine tending. If automated, this operation is mostly done by industrial robots. Manufacturing cells consisting of a machine tool and a robot are also known as robotized manufacturing cells (RMC). Generally, there are two solutions for the implementation of automated machine tending operations. One way is the integration of a work piece handling device into a machining center, the other way is to position a robot in front of the machine's work space [1][2].

Although the robot unit prices fell dramatically during the past 25 years, we still face an economical barrier to the automation of many manufacturing processes. There are two main reasons: On the one hand, the generation of robot programs for the handling of new work pieces is still laborious and time consuming, on the other hand, the lack of standardized interfaces for robot and machine tool controllers still prevents machine tending robots to be used efficiently with a range of different machines and systems. This fully applies to small and medium sized enterprises (SMEs) due to rather small and varying lot sizes.

Florian Pauker, Iman Ayatollahi, Burkhard Kittl
Institute for Production Engineering and Laser Technology,
Vienna University of Technology, Austria

The first problem has been addressed by robot manufacturers during recent years. Almost all robot manufacturers like ABB, KUKA or Fanuc have developed software packages making programming of machine tending solutions easier in order to reduce the effort for implementing robots in manufacturing cells.

The problem of insufficiently standardized communication interfaces in machine tending applications is an issue that still needs to be solved. Communication capabilities of machine tools and robots very often are restricted to simple I/O signals [3]. Reconfiguration of a robotized manufacturing cell therefore requires significant engineering expertise. OPC Unified Architecture (OPC UA), a new standard specification for interconnectivity in state-of-the-art industrial automation technology, is widely accepted as a promising enabler for the 4th industrial revolution. It provides a solution for moving information in secure reliable transactions between devices on the shop floor. Even with the introduction of OPC UA as a standard transport mechanism, one problem still to be solved is the introduction of a generally accepted semantic description of devices used in the manufacturing domain. Up to now communication issues in the automation environment have not affected a wide circle of interested parties. Machine users either operate individual machines, which do not interface with automation equipment or superordinate systems, or purchase preconfigured automated cells from machine manufacturers or system integrators. Communication issues often are neglected in the final purchasing decision. Machine and robot manufacturers are waiting for stable and mature communication standards before implementing them in their control systems, but standards cannot mature without feedback from application in practice.

This is why the Institute for Production Engineering and Laser Technology at Vienna University of Technology is working on information models and concepts for improving the interoperability of automation equipment in the manufacturing domain. A major goal of our efforts is the design of generic OPC UA information models for machine tools and handling devices in the context of robotized manufacturing cells, taking into account existing standards such as VDI 5600 or VDMA 34180.

As part of this research project we developed a prototype OPC UA server for an ABB IRB 120 Robot equipped with the Robotware Machine Tending option. The OPC UA server exposes the relevant objects (e.g. grippers) managed by the robot controller. These objects can own attributes for read and write access, methods that can be called (e.g. move commands), and trigger events.

Proc. of the Second Intl. Conf. on Advances In Mechanical and Robotics Engineering- AMRE 2014.
Copyright © Institute of Research Engineers and Doctors, USA .All rights reserved.
ISBN: 978-1-63248-031-6 doi: 10.15224/ 978-1-63248-031-6-155

A. *ABB machine tending solution*

ABB offers both a robot controller software option (RobotWare Machine Tending) and an add-in (Machine Tending Power Pac) for its offline programming software "ABB RobotStudio", which seamlessly work together. The combination of flexible controller software and an offline software tool allows easy configuration, programming and operation of machine tending tasks.

With the ABB Machine Tending Power Pac users can reduce their operational expenditure. With a few steps a virtual cell can be build up by adding common grippers, stations (e.g. machine tools, feeders or peripheral equipment) and parts from a library. This library is provided by ABB and can be extended with user defined objects [4].

The following flowchart (Figure 1) shows the recommended workflow for setting up a virtual cell in the offline programming software RobotStudio using the Machine Tending Power Pac (MTPP).

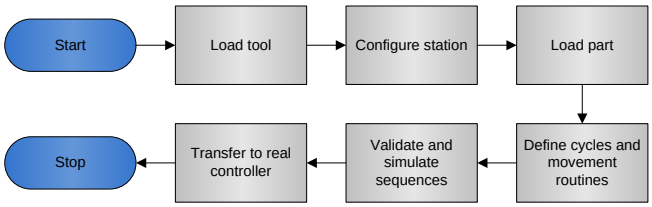

Figure 1 Workflow of ABB Machine Tending Power Pac

The first step is loading a robot tool from a library. This tool contains all necessary data like geometry, tool center point (TCP) and signals for tool actuation. Furthermore the machine tending stations are added to the cell, based on predefined types and RAPID templates. This leads to automatic generation of all corresponding movement routines and control signals necessary for each station. The work pieces (parts), previously generated by importing a CAD-file, are also loaded form a library.

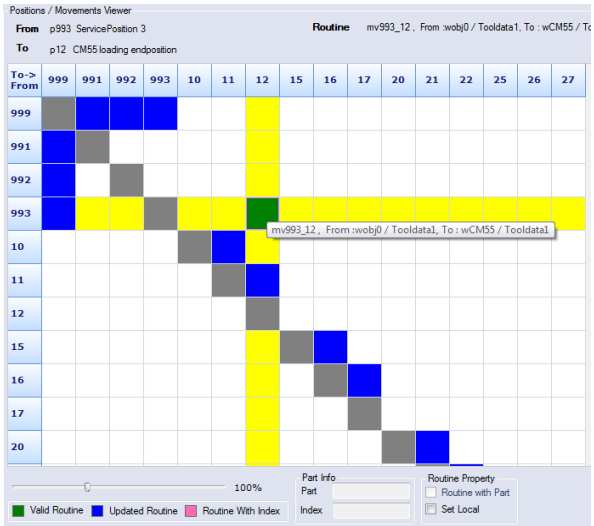

Figure 2 User interface (movement matrix) for teaching movements

The movement routines between the configured positions (home position, loading position, etc.) are partly generated automatically; missing ones should then be added manually. The MTPP provides an interface for easy programming of movement routines (Figure 2). Configured robot positions are listed in columns and rows of a matrix. Possible movements between the positions are presented as fields. The automatically generated routines are colored blue. By double-clicking on a white field a new routine based on a template is generated. After validating and testing the sequences in a simulation run the generated information, e.g. movement routines, gripper data and work piece definition is transferred to the real robot controller.

On the controller side, ABB's Machine Tending RobotWare in combination with a graphical Flex-Pendant interface allows easier access to robot functions and system peripherals. This interface is designed for less skilled operators to control most common machine tending tasks including production monitoring, as well as automatic program and part selection. Advanced users e.g. system integrators still have the possibility to use the conventional RAPID coding tools [5].

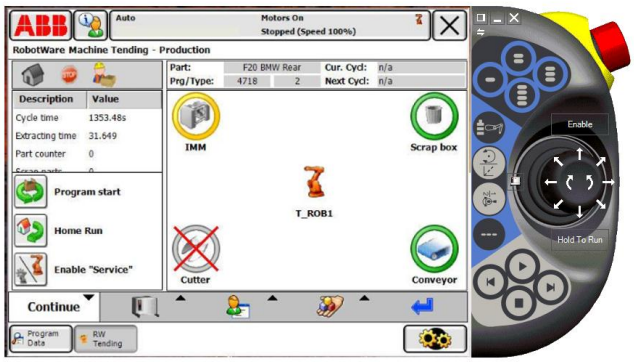

Figure 3 Flex-Pendant interface for machine tending

The user interface (Figure 3) is automatically generated by the MTPP and shows all cell components (robot, conveyor, etc.) and their operating status. The color of the circles indicates whether the station is ready for operating (green), busy (yellow) or out of order (red).

The combination of the RobotWare and the Power Pac allows programming, configuration and operation of machine tending applications. Applying these software solutions implies that the robot controller also orchestrates actions of the other involved devices (e.g. NC-machine) related to the machine tending operation. As communication interface, ABB uses the "Europmap 67" specifications that are based on binary I/O signals.

B. *OPC Unified Architecture*

Industry 4.0 is a project in the high-tech strategy of the German government, which targets better horizontal integration through value networks as well as vertical integration and networked manufacturing systems [6]. Making the visions of Industry 4.0 a reality requires an open and

Proc. of the Second Intl. Conf. on Advances In Mechanical and Robotics Engineering- AMRE 2014.
Copyright © Institute of Research Engineers and Doctors, USA .All rights reserved.
ISBN: 978-1-63248-031-6 doi: 10.15224/ 978-1-63248-031-6-155

standardized communication platform meeting the following requirements beside security aspects:

- Independence of manufacturer, industry sector, operating system and programming language
- Scalability for seamless networking of production system components from single sensors to MES and ERP applications
- Representation of any complex communication contents for modeling of virtual objects as representatives of real products and their production sequences

As the communication technology standard OPC Unified Architecture meets these requirements, it is seen as an enabling technology for Industry 4.0 [7]. It is the newest standard from the OPC Foundation for interconnectivity in state-of-the-art automation systems and was born out of the desire to create a platform independent solution for replacing previous COM-based OPC specifications. OPC UA can seamlessly be integrated in all layers of the automation pyramid, independent from the installed operating system or hardware (see Figure 4).

OPC UA enables information modeling using object-oriented techniques and defines a set of base types that can be extended by objects, references and data types defined by vendors, organizations or standardization committees. The idea is that OPC UA transport mechanisms specify data exchange, while the information models specify what information is exchanged [8].

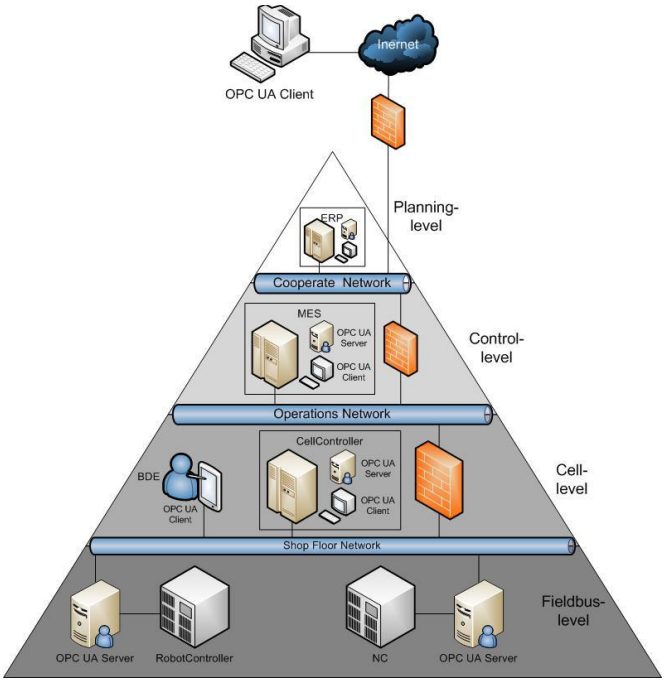

Figure 4 OPC UA as communication interface in the Automation Pyramid

At the Institute for Production Engineering and Laser Technology, we installed the prototype of a manufacturing cell consisting of an EMCO Concept NC machine and an ABB IRB 120 robot. Automated work piece handling and work piece machining are coordinated by a cell controller. Communication between cell controller, machine control and robot controller is executed through OPC UA interface. Details of the OPC UA server for the ABB robot are described in the following chapter.

II. OPC UA server Implementation

A. *Architecture*

Although, as pointed out before, the ABB machine tending solution is intended to coordinate the automated loading and unloading operations of the manufacturing cell by directly exchanging signals with the machine tool controller, we propose a different scenario. In this scenario all automated sequences are orchestrated by the cell controller communicating with robot and machine tool controller via OPC UA interface. We use the ABB machine tending solution for cell configuration and easy generation of movement routines. Figure 5 illustrates our approach.

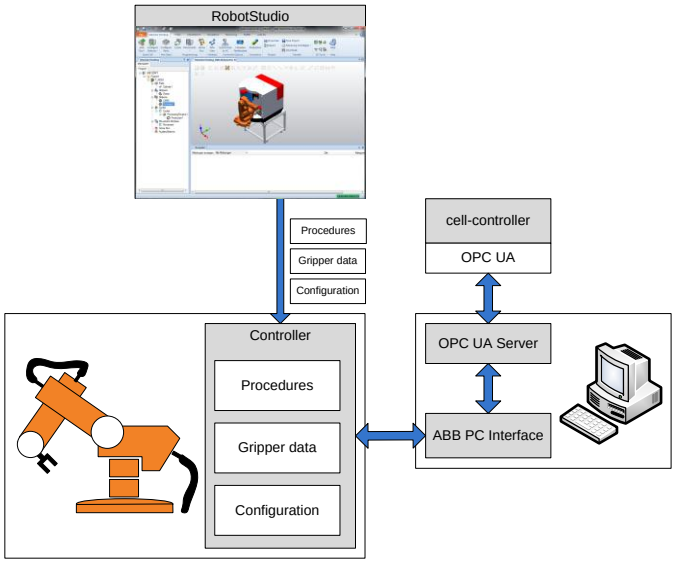

Figure 5 OPC UA server for ABB machine tending robots

The robot controller stores all information generated with the Machine Tending Power Pac. The motion sequences, required tool information, e.g. tool-data and the cell configuration are transferred from RobotStudio to the robot controller and are accessible via the ABB PC Interface, a communication interface between the robot and a PC in the same network. With this interface connections to either real or virtual robot controllers can be established.

The ABB PC Software Development Kit (PC SDK) enables a programmer to develop customized applications based on the PC interface [9] using included .Net libraries that facilitate development of C# or VB applications.

The OPC UA server was developed using a .Net SDK from Unified Automation. This SDK also provides a modeling tool for designing information models and generating a code

Proc. of the Second Intl. Conf. on Advances In Mechanical and Robotics Engineering- AMRE 2014.
Copyright © Institute of Research Engineers and Doctors, USA .All rights reserved.
ISBN: 978-1-63248-031-6 doi: 10.15224/ 978-1-63248-031-6-155

framework for server development. To establish a connection between the OPC UA server and the robot controller, methods provided by the ABB PC SDK were applied. As an example the SDK provides a method to read all variables of the type *"robtarget"*. So with one method call the server can read all robot positions. The information provided by the server is then accessible from other devices in the network that have OPC UA interfaces, in this case the cell-controller.

In Figure 6 the hierarchically structured address space of the OPC UA server is shown. With a standard OPC UA Client users can simply browse the address space. In this prototypic application the main object "ABB_IRB_120" contains the robot subsystems, each including all necessary variables and methods. These represent current values and functions of the robot.

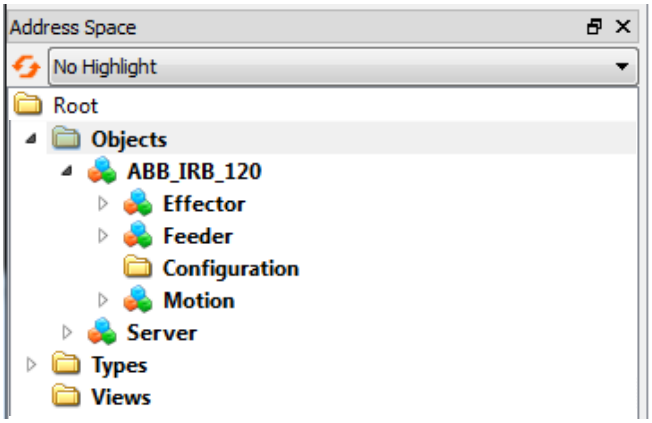

Figure 6 Screenshot of an OPC UA client showing the server's address space

The cell configuration generated with the MTTP is stored in the robot controller and mapped to nodes (objects, variables and methods) of the address space of the OPC UA server. The address space is generated dynamically, adapting the address space of the server to the given cell configuration.

B. *Dynamic object generation in the address space*

The *Effector* Object contains information and methods for the robot-tool. By invoking the method „ChangeTool" the robot is instructed to change its effector. As the effectors are binary coded, the robot controller knows which tool actually is in use. Whenever the effector is changed, the server reads the binary code, deletes the old effector object from the address space and adds an instance of the new effector from its objecttype (e.g. GripperType) to the address space, as shown in Figure 7. Subsequently the server generates an event, which requests connected clients to rebrowse the address space.

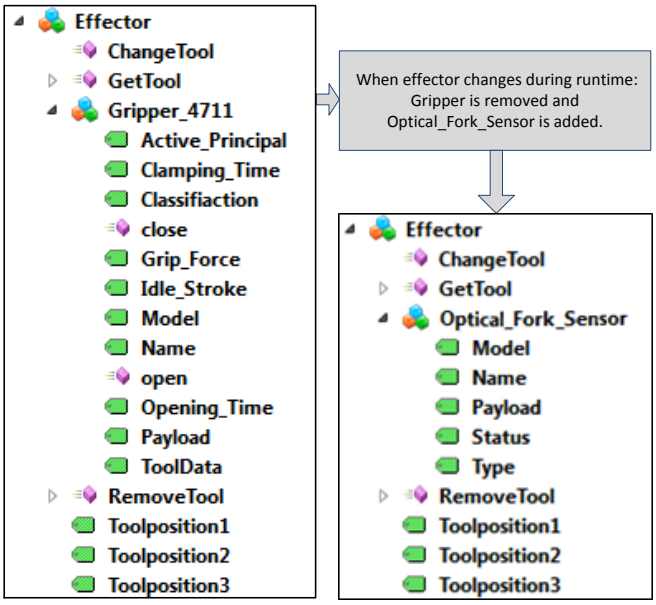

Figure 7 Effector object with Gripper object (left) and Optical_Fork_Sensor object (right)

C. *Robot program execution*

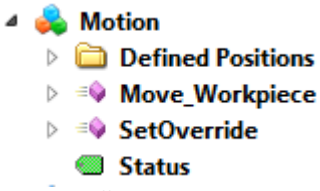

Figure 8 Motion object with offered OPC UA methods

All robot positions, defined with the Machine Tending Power Pac, are stored in a folder (Defined Positions). Calling the method "Move_Workpiece" with the input arguments start and end position, prompts the server to check all conditions for safe movement. Figure 9 shows the flow chart of this method execution. Whenever the method is called, the OPC UA server checks whether a robot program is running and so prevents simultaneous method execution.

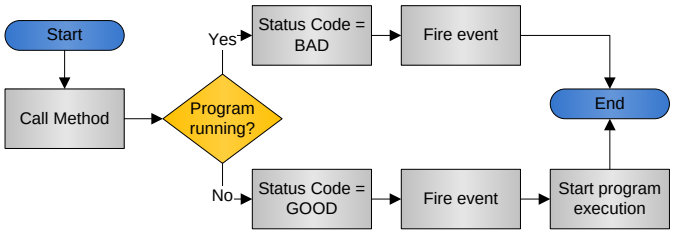

Figure 9 Flow chart of method execution for "Move_Workpiece"

Proc. of the Second Intl. Conf. on Advances In Mechanical and Robotics Engineering- AMRE 2014.
Copyright © Institute of Research Engineers and Doctors, USA .All rights reserved.
ISBN: 978-1-63248-031-6 doi: 10.15224/ 978-1-63248-031-6-155

If no robot program is running the server returns a "GOOD" StatusCode, fires an event and starts program execution. The value of the Status variable is then also changed to "running".

D. *Event based value updates*

Programming the OPC UA server in .NET also allows applying .NET-events as OPC UA-events. In combination with events provided by the PC SDK the OPC UA server can update values of UA variables event-based. It's also possible to generate OPC UA alarms with such events.

```
void fireevent(string message)
       {
            // create the event.
            GenericEvent motionevent = new
GenericEvent(OPC4IRB_Intertool.TestServerManager.nm.Server.FilterManager);
            // initializ base event type fields
            motionevent.Initialize(
            null, // EventId created by SDK if null
            UnifiedAutomation.UaBase.ObjectTypeIds.BaseEventType, // EventType
            UnifiedAutomation.UaBase.ObjectIds.Server, // SourceNode
            "Motion", // SourceName
            EventSeverity.High, // Severity
            message); // Message
            // report the event.
            OPC4IRB_Intertool.TestServerManager.nm.ReportEvent(motionevent.SourceNode,
motionevent);
       }
```

Figure 10 Code snippet Event

In this application the *"GenericEvent"* type for all events is used. Each event transmits the information which node (object, method or variable) generates the event. Additionally a message with more detailed information is delivered.

III. Conclusion

As we already mentioned before, robotized manufacturing cells still cannot be used efficiently in a high mix low volume environment. One of the main reasons is the lack of standardized interfaces for machine tool and robot controllers. Although OPC Unified Architecture seems to be a promising communication standard, we still miss a generally accepted semantic description of manufacturing cell components. In this paper we present a prototype installation of an OPC UA server for an ABB machine tending robot. The server accesses the cell configuration data in the robot controller generated by the ABB Machine Tending Power Pac and represents objects such as grippers and robot targets in terms of variables and methods to clients (e.g. the cell controller). Our approach has been tested in a simple use case, where a cell controller orchestrates activities of robot and machine tool by calling the appropriate methods presented by the respective OPC UA servers. Future research must be focused on the development of universal OPC UA information models for robots and machine tools in a machine tending scenario able to cover the requirements of common cell configurations and peripheral equipment such as zero-point clamping systems found on the market.

References

[1] "AUTOMATICA 2014: Roboter für die Metallbearbeitung – Breites Angebot für alle Einsatzfälle," 22-Jul-2014. [Online]. Available: http://automatica-munich.com/link/de/27681677.

[2] Mikael Gardh, "Machine Tool Tending." 18-Nov-2006.

[3] B. Solvang, G. Sziebig, and P. Korondi, "Multilevel control of flexible manufacturing systems," 2008, pp. 785–790.

[4] ABB Robotics, "ABB Machine Tending Software Brochure - Easy, flexible and trouble-free system for robotic machine tending." 19-Dec-2012.

[5] ABB Robotics, "Application manual - RobotWare Machine Tending." 07-Mar-2013.

[6] H. Kagermann, W. Wahlster, and J. Helbig, "Abschlussbericht des Arbeitskreises Industrie 4.0," Berlin, finalreport, Oct. 2012.

[7] OPC Foundation, "OPC Unified Architecture - Pioneer of the 4th industrial (r)evolution." .

[8] W. Mahnke, S.-H. Leitner, and M. Damm, OPC Unified Architecture, 1st ed. Springer Verlag, 2009.

[9] ABB, "Application manual PC SDK." 15-Mar-2012.

About Author (s):

Florian Pauker is researcher at the Institute for Production Engineering and Laser Technology at Vienna University of Technology. There he is responsible for robotized manufacturing, especially new concepts of connecting robots with machine tools. In his diploma thesis a first try for implementing plug & produce in RMCs was given

Iman Ayatollahi is researcher at the Institute for Production Engineering and Laser Technology at Vienna University of Technology. Currently he is working on development of information models for machine tools and peripherals.

Burkhard Kittl is associate professor at the Institute for Production Engineering and Laser Technology at Vienna University of Technology. His main research topics are Manufacturing Execution Systems based on Service Oriented Architectures and their integration with monitoring and control level.

View publication stats