MULTI AGENT BASED CONTROL ARCHITECTURES

Solmaz Mansour Fallah[a]

[a]TU Wien, Institute for Production Engineering and Laser Technology, Vienna 1060, Austria

Abstract

The pressure of global competition has led the urge of highly individualizes products to sustain in the market. This entailed a rise of information technologies in area of discrete manufacturing, which has led to Computer Integrated Manufacturing (CIM) and later to Reconfigurable Manufacturing Systems (RMS). Both paradigms struggle with stiff hierarchical control architectures, and a lack of flexibility. To overcome those limitations, the aim is to integrate the concepts of Service Oriented Architecture (SOA) into the factory automation domain. By elaborating a way to induct the paradigm of SOA in Manufacturing Execution System (MES), a supporting framework will be established. This will enable an autonomous model for multi agent based control architecture of a demonstration cell. This proposal aims to work out those synergies and to give a short envision of the state of the art, in order to expound the authors previous approach and an outlook. The previous approach was built around a demonstrator manufacturing cell, where an OPC UA server has been developed for the machine control. Through the capability of evolving method into the information model, the opportunity of sequencing rudimentary services to higher services processes occurred.

The outlook will show how we plan to merge those higher services to a service oriented Manufacturing Execution System, by taking the guideline VDI 5600 into consideration.

Keywords: MES; SOA; Multi Agents; VDI 5600; Cyber Physical Systems; Industry 4.0

This Publication has to be referred as: Mansour Fallah, S[olmaz] (2016). Multi Agent based Control Architectures, Proceedings of the 26th DAAAM International Symposium, pp.1166-1170, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-07-5, ISSN 1726-9679, Vienna, Austria
DOI: 10.2507/26th.daaam.proceedings.164

1. Introduction

Due to the rising pressure of global competition western industries need to adapt their production to meet the demand. Customers expect individualized product to fit their needs, while they demand low prices as common to mass production. The automotive industry for example, highly individualizes products, as they can be customized online. To manage conflicting goals such as low cost production and high product variation, an adaptive and efficient manufacturing planning and control will be necessary. In order to act to this exigency, the German Federal Government initiated the "Industrie 4.0" project. It urges the development of the fourth industrial revolution, the "smart factory". The model of a smart factory will be based on the integration and cooperation of two technologies, "Cyber-Physical-Systems" (CPS) and "Internet of Things" (IoT). CPS are basically network embedded technical systems, which can include physical structures, and will be capable of autonomously exchanging information, triggering actions and controlling each other independently [1].

In recent decades, a trend of enhancing Computer Integrated Manufacturing (CIM) emerged. CIM systems bring about advantages, such as control organization and data acquisition organizations, but they also struggle with a stiff and complex hierarchy. For those reasons, a continuous reorientation of CIM for Reconfigurable Manufacturing Systems (RMS) became indispensable. RM systems allow for rapid changes of their structure, as well as of the embedded hardware and software. Nevertheless, they are challenged by the high efforts that are necessary to implement configuration changes [2]. To overcome those challenges, we aim to integrate the concepts of Service Oriented Architecture (SOA) into the factory automation domain, to ease programming and integration of automation components. By elaborating a way to induct the paradigm of SOA in Manufacturing Execution System (MES), a supporting framework will be established. This will enable an autonomous model for a multi agent based control architecture of a demonstration cell.

First part of this paper will be a short introduction to MES and SOA, only to work out synergies. The second part will give a brief envision of the state of the art, in order to define our scope. In the next part we will expound our approach and reference primary work and already gained solutions. The last part of this paper will contain our outlook on this topic.

2. MES, SOA and their compatibility

2.1. MES

The term Manufacturing Execution System (MES) is not defined consistently, due to its evolved structure. The most common definition was established by the Manufacturing Enterprise Solution Association International (MESA International). It defines MES as a complementary information system for manufacturing processes, ERP, and control system [3]. Through the mapping of the actual existing situation of production, it enables advantages as detail planning, efficient resource planning, continuous control and data acquisition. Not all areas of MES are new, for example the production data acquisition, but those different paradigms were developed separately. The rise of CIM gave the initial impulse to recapitulate those different areas and to complement them to MES. The need for MES brought several standards and guidelines, as for example the VDI 5600. Its task-oriented and multivendor-capable description of MES and their application field captures the existing concept of MESA and complements it. VDI 5600 was published by the 2003 founded VDI-Competence Group IT (German KfIT). It focuses on representation and adaption of MES, unlike other standards and guidelines (NE 94, ANSI/ISA 95, IEC 62264), which focus on the realization [4].

2.2. SOA

This paradigm aims to offer a catalogue of atomic services, which are self-describing, in order to run business processes. Basic services can be orchestrated to higher services, by triggering or including other basic services. The main idea during development was, that business processes run IT- and control system, instead of being limited and dictated by them. This paradigm enables more flexibility [5]. The conceptual understanding of Service-Oriented Architecture (SOA) varies in literature, only a few conformities establish, autonomy, as an example to be given. It emphasizes independency between different basic services. Another conformity is the platform independence, as services are described in text-based formats, which also supports availability. Latter conformity exposes all services in public registries. Basic services were defined as self-contained functions in the course of encapsulation, in order to get rid of unnecessary details. Composing those basic services enables orchestration of higher level services [6].

2.3. Compatibility

Several organizations have aimed to implement a Service-Oriented Architecture approach into the automation domain. The purpose is to bring a holonic system modelling at different levels of abstraction together, with process detailing and implementation at the execution level [7]. By elaborating a way to induct the paradigm of SOA in Manufacturing Execution System (MES), a supporting framework will be established which enables autonomous model of a multi agent based control architecture. The SOA frame will be built, by using the task-oriented MES description, VDI 5600, to map MES-functions and their application potentials.

3. State of Art

Recently Agents experienced a significant upturn, due to their nature to fit into decentralized control structure. In this section we will present two different approaches of agent based service architecture in the production domain, to describe the actual state of art, and to give a short overview of two different approaches in this area.

3.1 MyJoghurt demonstrator factory

The first approach is the so called MyJoghurt demonstrator factory, from Birgit Vogel-Heuser and her team. This demonstrator shall realize interlinking of territorial distributed production facilities, by using a loose coupling paradigm based on a Multi Agent System [8].

A provider, who sells the final product with his label, is represented in his production alliance (union of supplier factories) through his provider agent. Which supplier factory will get an assignment to provide its production services, will be decided by the provider agent. It is negotiating through the specific Yoghurt-Production-Protocol (JPP) with different coupling agents, considering their offers from. Each supplier factory consists of its specific internal structure, including MES, PLC, visualisation and coupling agent, which represents the factory in the production alliance. Facilities, which offer the same output, will compete against each other by offering the best possible price and delivery time. The core of this approach is the JPP protocol used to standardise the interface between coupling agent and provider agent and to implement a service oriented text-based communication. It is based on the Transmission Control Protocol (TCP) and has three independent layers inside. The basic layer, which has been fully implemented by now, includes header definition, the volume and structure of the protocol. The service layer provides data, due to inquiring production costs and delivery time. The top layer is the data layer, in which the semantic model is embedded [9]. This specific protocol was designed for the domain of yoghurt production. In order to use the structure of this protocol in other domains efforts in all three layers will be needed.

Although the communication between the provider agent and the coupling agent is defined through JPP, the communication between coupling agent and MES or PLC has been left to the factory choice or capability.

The main task of the multi agent system and their decision making is to choose a suitable (price/delivery time) factory to produce and to initiate the production. Considering those actions and the layer they are embedded, this demonstration factory is using its Multi Agent System in the ERP.

3.2 GRACE Project

The second approach to be described aims to implement autonomous and self-reconfiguring agents into the production domain to improve shop floor efficiency and maintenance. The EU FP7 research project of the team around Paulo Leitão, called GRACE, develops MAS for process control of a washing machine production line. This concept has been tested in an experimental environment [10]. A vision inspection station designed as an agent will be implement into the GRACE Multi Agent platform to communicate and exchange data with it, in order to optimize its operations.

The GRACE MAS platform consists of several different agents, we introduce a short selection: Product Type Agent (PTA), which represents products or parts and contains the product knowledge, like product structure and needed manufacturing schedule. A Product Agent (PA), which manages the production of product instances in the production line, has knowledge about the process plan, interacts with Resource Agents for process and quality control. An Independent Meta Agent (IMA) enables the influence of global supervisory control, optimized planning and decision-making. Considering feedback from previous quality control tests and advices from the IMA, the PA chooses the best algorithm or parameters for its decisions for the next quality control operation. A new customized test plan will be generated and communicated to the quality control station (the physical layer of quality control agent). The self-optimization and self-adaptation of the quality control agent is possible, and control loops in the production line can be closed to improve the quality of the manufactured appliances and the efficiency of the entire production.

Experimental tests have verified that the proposed implementation of the quality control agent enhances efficiency of the production line due to their ability to adapt the operations of the quality control station [10].

This approach has similarities with our approach. It aims to break the stiff hierarchical control structure in MES, by bringing self-optimizing and self-adapting agents into the shop floor. Those agents do not have to be bound to physical layer, but are most likely.

4. Previous Approach

Our demonstrator manufacturing cell consists of a cell controller, an EMCO Concept Turn 55 machine tool, and an ABB IRB 120 industrial robot. An OPC UA server has been developed to enable a nonproprietary remote control of the machine. The communication between OPC UA server and numerical control of the machine tool is based on a proprietary DNC protocol. An information model, as a conceptual representation of relationships, constraints, rules, and operations had to be developed. Our OPC UA server exposes objects, which typically represent entities in the real-world. Those objects can contain variables and methods, which are used to invoke actions on the object and events. Due to this possibility of OPC UA to introduce methods, this solution seems to be an appropriate fit in area of discrete manufacturing [11].

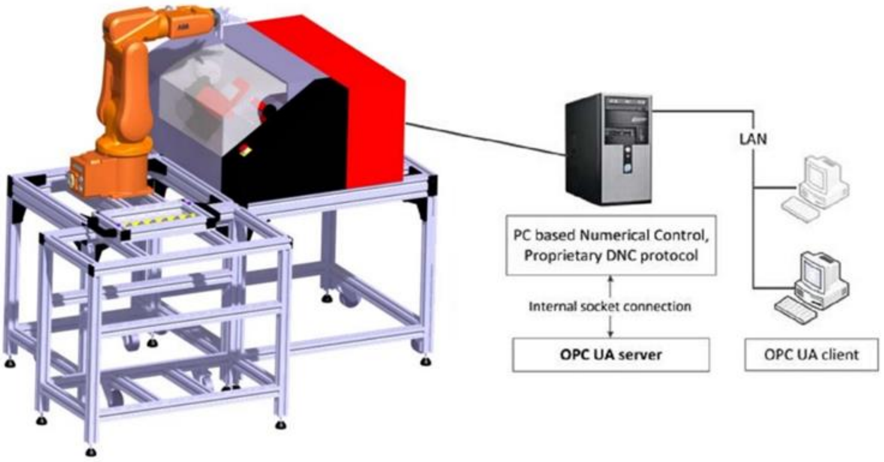


Figure 1: The robotized manufacturing cell at IFT [11]

“MachineType” as a main object type enfolds instances of automated machine tool components, like loading door or clamping device. This object oriented programming, enables dynamic instantiation during server runtime, e.g. removing tools from the tool magazine. An OPC UA client with appropriate rights is able to access the information model, and therefore to read values of variables or call methods [12].

As the automated manufacturing group of the IFT published in 2015 [13], chosen methods can be added to a sequence as an action. If the line represents a method, the method is called and after receiving the return parameter the client handles the next line. If the line is a transition condition the value of the respective OPC UA variable is monitored until the specified condition is met.

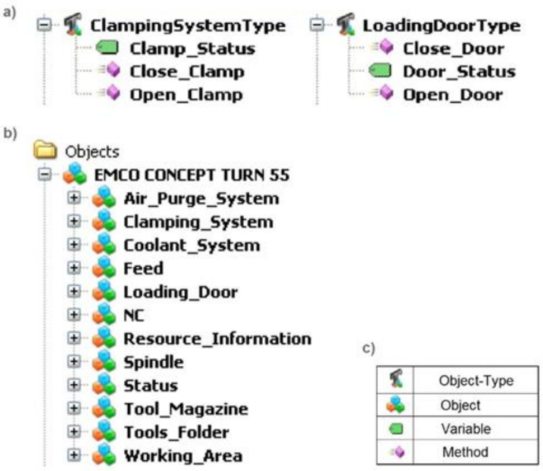
The cell controller collects data from the machine tool and is able to remotely control the machine tool. The demonstrator manufacturing cell has been tested [11].

5. Outlook

As the previous projects have shown Multi-Agents-Systems seem to be a proper way to implement a certain level of flexibility and autonomy. An SOA implemented MES will have a sufficient flexibility to transact decentralized control structures, in which a range of service providers could be integrated in a loosely coupled way [14].

We will induct the paradigm of SOA in MES, by using the guideline VDI 5600 to develop a proper catalogue of services for the metal-cutting production. We will elaborate necessary MES-functions, as data acquisitions, offered services, and control functions, etc., by triggering and including basic equipment services. By doing so our services will be provided by basic equipment functions that execute and monitor operations.

Those rudimentary services can already be composed to higher services (sequences), which will be able to control the production process. Examples for such rudimentary operation service at the equipment level are openDoor(),openClamp(), startNC() etc. In order to transact higher level operation services, we implemented their semantics by orchestrating rudimentary services to coarse-grained operations. startMachineLoadingProcess() for example, includes composed services like openDoor(), openClamp(), movePart(), etc.


a)

ClampingSystemType
- Clamp_Status
- Close_Clamp
- Open_Clamp

LoadingDoorType
- Close_Door
- Door_Status
- Open_Door

b)

Objects
- EMCO CONCEPT TURN 55
  - Air_Purge_System
  - Clamping_System
  - Coolant_System
  - Feed
  - Loading_Door
  - NC
  - Resource_Information
  - Spindle
  - Status
  - Tool_Magazine
  - Tools_Folder
  - Working_Area

c)

	Object-Type
	Object
	Variable
	Method

Figure 2: Information Modell: a) Object-types of an auxiliary driver, b) The machine object and sub-objects, c) Legend [15]

To veer off the usual rigid and inflexible hierarchy, we will provide decentralized control structures through the orchestration of those services. This orchestration will not be defined as fixed actions, which were mapped to specific devices, but rather as a course including decision nodes. Decision making will not only be based on device or controller information, but also on higher level service entities. The decision nodes shall not appear as fixed solution paths, but rather as an intersection of interest between two or more entities.

This approach will grant a maximum of flexibility and changeability, which enables the production flow to a quick response if necessary.

To realize those decision-making entities, a Multi-Agent System (MAS) with an interface to the SOA implemented MES structure can be used. Baker [16] defined an agent as a self-directed software object. We define an agent as an autonomous, self-deciding software object, which can, but does not have to be bound to a physical layer (CPS). It shall be an object with its own value system, capable of communicating with other agents, and acting continuously upon its own initiative [17]. For classifying and shaping a sufficient environment, our catalogue of services seem to be a proper directive to encapsulate the different field of actions.

We will implement that orchestration paradigm in our demonstrator cell, which we introduced before. We also aim to use a common communication interface standard, OPC UA, as middleware, seems sufficient to integrate agents into the physical layer of the equipment. Additionally it defines generic services and seems to be capable of serving as a SOA-based middle ware for a Multi-Agent System [18].

Our expected results are to develop a domain model for robotized manufacturing cells and to design a control architecture by integrating MAS and a SOA implemented MES based on OPC UA as a middle ware.

6. References

[1] Arbeitskreis Industrie 4.0, Umsetzungsempfehlungen für das Zukunftsprojekt Industrie 4.0. Abschlußbericht, April 2013

[2] A. Maffei, K. Dencker, M. Bjelkemyr, M. Onori: From Flexibility to Evolvability: ways to achieve self-reconfigurability and full-autonomy. Robot Control, 74-79 (2009)

[3] Kurbel K.: Produktionsplanung und –steuerung im Enterprise Resource Planning und Supply Chain Management. Oldenbourg Verlag, 264-266, (2005)

[4] Guideline, VDI: 5600 (2007) Fertigungsmanagementsysteme, Manufacturing Execution System (MES). Blatt 1. p.5, (2013)

[5] MESA International, IBM Corporation Capgemini Co.: SOA in Manufacturing Guidebook, white paper 27, 24-29 (2008)

[6] Ribeiro L., Barata J., Mendes P.:MAS and SOA: complementary automation paradigms, Innovation in manufacturing networks. Springer US, 259-268 (2008)

[7] Pfadenhauer K., Kittl B., Dustdar S.: Challenges and solutions for model driven web service composition. Enabling Technologies: Infrastructure for Collaborative Enterprise, 2005. 14th IEEE International Workshops on, 126-131 (2005)

[8] Mayer F., Pantförder D.,Diedrich C.,Vogel-Heuser B.: Deutschlandweiter I4. 0-Demonstrator-Technisches Konzept und Implementierung, Technischer Report (2013)

[9] Bauernhansl T., Ten Hompel M., Vogel-Heuser B.: Industrie 4.0 in Produktion, Automatisierung und Logistik, Springer-Verlag (2014)

[10] Stroppa L., Rodrigues N., Leitão P., Paone N.: Quality control agents for adaptive visual inspection in production lines, IECON 2012-38th Annual Conference on IEEE Industrial Electronics Society 4354-4359 (2012)

[11] Ayatollahi I., Kittl B., Pauker F., Hackhofer M.: Prototype OPC UA Server for Remote Control of Machine Tools, Proceedings of International Conference on Innovative Technologies 73-76 (2013)

[12] Pauker, F., Ayatollahi, I., & Kittl, B.: Service Orchestration for Flexible Manufacturing Systems using Sequential Functional Charts and OPC UA. Dubrovnik, 9, 11-09(2015)

[13] Pauker, F., Ayatollahi, I., & Kittl, B.: OPC UA for machine tending industrial robots-Prototypic development of an OPC UA server for ABB industrial robots. In Second International Conference on Advances In Mechanical and Robotics Engineering-AMRE (2014)

[14] Pfadenhauer K., Kittl B., Dustdar S., Levy D.: Shop floor information management and SOA. Business Process Management Workshops, 237-248 (2006)

[15] Ayatollahi, I..: Development of an OPC UA Server for a numerically controled machine tool, Master Thesis, TU Wien, 20-22 (2014)

[16] Baker A.D.: A survey of factory control algorithms that can be implemented in a multi-agent heterarchy, Journal of Manufacturing Systems 17, 297-320 (1998)

[17] Csáji B.C., Monostori L., Kádár B.: Reinforcement learning in a distributed market-based production control system, Advanced Engineering Informatics, 279-288 (2006)

[18] Henßen R., Schleipen M.: Interoperability between OPC UA and AutomationML, Procedia CIRP 297-304 (2014)