OPC UA Based ERP Agents: Enabling Scalable Communication Solutions in Heterogeneous Automation Environments

Max Hoffmann(✉), Tobias Meisen, and Sabina Jeschke

Institute of Information Management in Mechanical Engineering,
RWTH Aachen University, Dennewartstr. 27, 52068 Aachen, Germany
{max.hoffmann,tobias.meisen,sabina.jeschke}@ima.rwth-aachen.de
http://www.ima.rwth-aachen.de/en/home.html

Abstract. This work contributes to a technology stack that pursues the goal of integrating intelligent entities in a production environment by means of communication technologies based on scalable interfaces supporting semantic modeling. The proposed architecture is realized based on a model-driven interconnection of multi-agent systems with OPC Unified Architecture. The integration of these technologies enables a usage of intelligent mechanisms within modern production sites while ensuring semantic integrity during all communication processes and compliance to essential security standards. The goal of this work is to enable a autonomous, reactive production by means of intelligent communication in autonomous systems. By making use of a model-based representation of intelligent software agents, an integration of cyber-physical systems with products and production units in manufacturing systems can be realized. The integrability of these multi-agent systems with high-level applications in terms of generic vertical interoperability is shown by means of seamless information exchange with an ERP system.

Keywords: Interoperability · Vertical integration · OPC UA · Semantic interface standards · Multi-agent systems · Enterprise resource planning

1 Introduction

One major topic of current research on modern production is dedicated to an intelligent utilization of resources [8]. With regard to industrial production, on the one hand these resources are characterized by the availability of manufacturing entities such as machines or engineering plants and on the other hand by means of human resources. The machine-related components of production plants are nowadays characterized by a high degree of automation, while the human-related tasks are moving from manual activities towards decision making. However, current developments show that autonomous decision making and intelligent behavior of manufacturing units can be also achieved by smart embedded autonomous systems working together in a cooperative manner [2].

Y. Demazeau et al. (Eds.): PAAMS 2017, LNAI 10349, pp. 120–131, 2017.
DOI: 10.1007/978-3-319-59930-4_10

In order to utilize the full potential of human decision making, automated legacy systems and autonomously acting intelligent systems, an efficient cooperation between these different players is required. The goal of this cooperation is to reach a holistic optimization of current and future processes in manufacturing systems by taking into account highly granular data from the lowest level of a production plant (shop floor) and combining them with high-level logic of planning systems, e.g. enterprise resource planning (ERP). The key for incorporating these highly heterogeneous systems is to reach a communication that suits low-level automated systems, intelligent entities as well as high-level systems that interact directly with human beings at the same time [3].

In the ideal scenario, an integration of all systems enabled by scalable communication leads to an omnipresent availability of information from the production. One major goal is to reach an optimal provision of data from the field level with the purpose of serving valuable information to human decision makers. To reach this target, raw data from the manufacturing level has to be preprocessed and appropriately visualized in order to serve as a suitable basis for interpretable information. Another major interest focuses on the utilization of data directly on the shop floor based on computer-aided tools [4]. These smart embedded systems usually focus on methods of artificial intelligence that go beyond simple algorithmic correlations and calculation rules that are already present within current control units and programmable logic controllers (PLC).

Although both of these strategies – integration of information from the shop floor and autonomous processing of this data – play an important role in production, pursuing only one of those will not exploit the potential of optimizing the entire process. Thus, this work is about combining both strategy goals – enabling profound human decisions by means of an interconnection between different systems and utilizing the potential of smart autonomous systems in the field. The interaction and vertical integration is realized through an information modeling approach that allows for scalable machine to machine communication through semantic model definitions for multi-agent systems (MAS) in the field.

The proposed model based representation of agents covers the modeling of agent entities and their communication with other systems of the manufacturing infrastructure. The state-of-the-art section covers basics about the communication in modern production sites and MAS. Section 3 describes a software stack for modeling MAS using a scalable approach. In Sect. 4 the scalability of the approach together with higher systems of the enterprise planning layer is validated by means of an extended ERP agent communication stack and use-case.

2 State of the Art

2.1 Communication in Modern Manufacturing Environments

Optimization strategies in automation systems depend on the availability of up-to-date information that is collected through various devices, such as sensors or other data acquisition tools. However, the collection, aggregation and automated interpretation of the available information poses huge challenges to current information management systems as production data from different sources is usually represented in various forms. Especially with regard to automated systems some distinct characteristics of available information can be pointed out:

- Production data collected in automation systems is naturally characterized by a strong degree of heterogeneity. The information does not only differ due to syntactical variance of data structures, but also in the manner different semantical concepts are used to represent the information.
- Modern production sites are comprised of a vast number of complex systems structured by means of hierarchically organized architectures, thus posing high challenges regarding vertical interoperability throughout these systems.

Hence, an integration of consistent interface technologies that connects all of these complex systems and subsystems with higher levels of the production planning and organization is difficult to achieve (see Fig. 1).

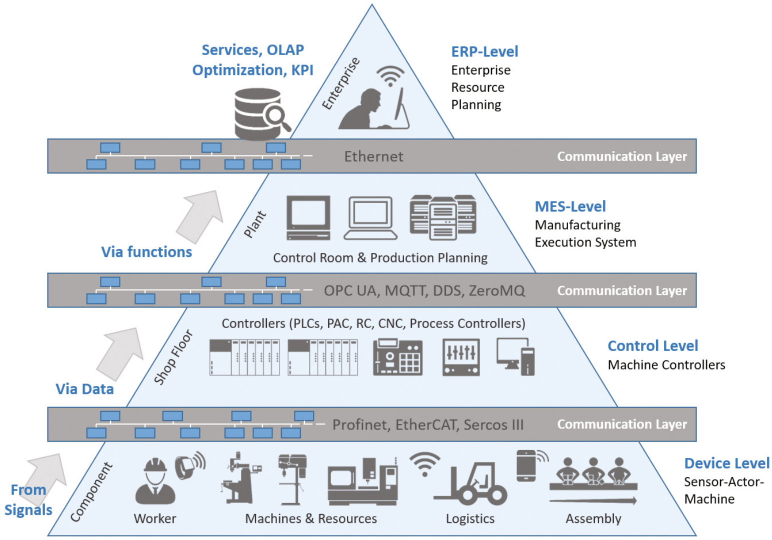

Fig. 1. Communication systems throughout the automation pyramid [1].

Seamless, integrative communication is generally only realizable in horizontal layers of automated processes, i.e. between programmable logic controllers or within closed control loops. Information to higher levels is mostly exchanged only in form of aggregated or condensed information that does not appropriately represent the complexity and granularity of the underlying production data.

In order to enable full availability of precise production data within higher systems for flexible process adaptation, the information from the shop floor has to be appropriately preprocessed and semantically annotated in order to fit the information management requirements on higher systems [11]. A promising communication interface standard that fits these requirements is OPC UA.

2.2 OPC UA – Scalable Interface Standard for Automated Systems

OPC UA has been derived from an initiative of a few major automation companies to a de-facto standard in automated industrial environments [5]. The OPC UA specification does not attempt to define another proprietary communication protocol, but rather introduces a metamodel that defines how information has to be represented in order to be integrable with information from other systems. For this purpose, OPC UA proposes an information modeling stack (Fig. 2) that allows for scalable extension of existing standards in order to fit the needs of each application or device that attempts to communicate through OPC UA.

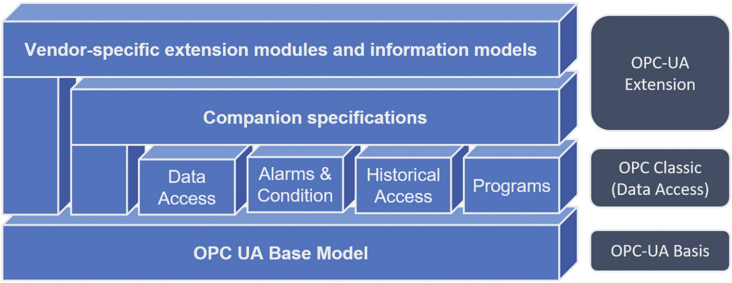

Fig. 2. OPC UA metamodel and information model stack

At the bottom of the information modeling stack the *OPC UA Base Model* is located. On top of the base model, additional models from former versions of the OPC standard preserve legacy system compatibility of OPC UA. The specifications of other organizations expand the standard by additional information models that comply to existing specifications, e.g. such as the IEC 61131-3 standard for PLC programming or AutomationML just to name two. On top of the model stack vendor specific namespaces can be defined., e.g. to facilitate an integration of common programmable logic controllers or similar pervasive device families. This approach pursued by OPC UA enables scalability on top of existing standardizations. The underlying design pattern can be used for a object-oriented modeling of arbitrary components, such as intelligent agents.

2.3 Enabling Cyber-Physical Systems in Production Through MAS

In order to cope with the growing complexity of modern production systems, ideas to facilitate optimization procedures based on intelligent, autonomous systems have been emerged in the last couple of years. In the matter of fact, the extensive increase of technical systems in automation environments brings centralized systems to its limits [10]. A key concept that is able to tackle these arising challenges consists in the introduction of cyber-physical systems (CPS).

The CPS idea is based upon the concept that all physical actions can be linked to a digital representation. Through this approach, it is possible to incorporate physical procedures with a *digital twin* that reflects real world processes with their digital counterpart [9]. An intelligent software agent as applied within multi-agent systems represents such a cyber-physical system. According to the definition of these intelligent entities, software agents provide capabilities to independently interact with their surrounding environment and performing autonomous actions while cooperating with other systems [7].

In accordance to CPS, agents are capable of sensing their direct environment and thus transporting physical actions into a digital context. In order to integrate these agents with other systems of the automation pyramid, this contextual information needs to be integrated with the semantics and information representation of the system architecture as demonstrated in the next section.

3 Agent OPC UA – A Scalable Approach for Integrating Multi-agent Systems into Real Production Sites

The OPC UA metamodel for information modeling is capable of incorporating any kind of object-oriented structure into its model definitions. In the same manner as companion specifications on the third and fourth layer of the OPC UA metamodel stack, an integration of complex concepts such as an object-oriented representation of MAS is also realizable by means of such approach. This section describes further developments of such specification that incorporates the representation and communication flow between agents and with high-level enterprise systems based on previous works of the authors [6].

3.1 Information Modeling for CPS

Similar to other smart embedded devices, intelligent software agents can be interpreted as some sort of cyber-physical system (CPS) as they are located on the interface of the physical process and its digital representation. The modeling of such entities is strongly inspired by the Internet of Things (IoT), which is also characterized by the formation of agile (ad-hoc) networks cooperating in agile infrastructures. Such open, dynamic and autonomous network compounds are only realizable through flexible information transport approaches that do not rely on central management instances, e.g. based on fixed servers, which mediate the communication processes between clients. The goal of the targeted network structures is to enable client as well as server functionalities on each communicating instance as vital part of the production network.

The presented approach aims at reaching such kind of generic interoperability between agents by enabling scalable communication solutions being part of a MAS metamodel. The proposed architecture is presented in the following section.

3.2 Semantic Integration of Intelligent Software Agents

The model representation of software agents based on OPC UA is performed by extending the existing OPC UA metamodel structure by an object-oriented mapping of the agents and their according communication skills (Fig. 3).

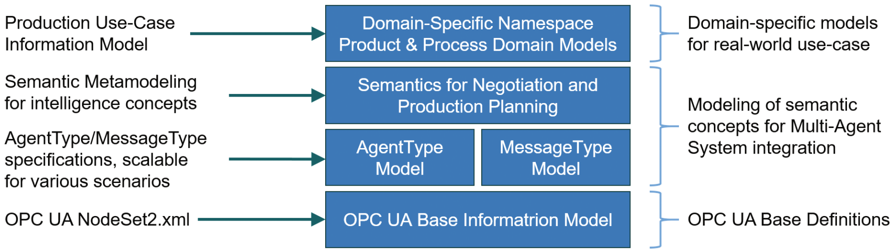

Fig. 3. Metamodel stack of OPC UA based multi-agent systems

On top of the base model, the *AgentType* and *MessageType* object models are located. The *AgentType* specification contains basic properties of an agent, e.g. the *capabilities* of an agent and its interactions facilities in terms of *environmental perception* through sensors and other data acquisition techniques.

The *MessageType* specification is located on the same semantic level as the *AgentType* and defines the inner design of messages that are exchanged between agents. The defined types reflect the different purposes of messages, e.g. for information exchange or for negotiation with other agents. Both metamodel specifications, the *AgentType* and the *MessageType* model, are designed by means of object-oriented modeling paradigms and comply with the basic requirements of OPC UA. This way, the information models are scalable in terms of new agent types or semantic enrichment of the message payload. Figure 4 shows the most important parts of the information model focusing on the *MessageType*.

The *BaseObjectType* is part of the base model and is located on the top representing the entry point for the information model. The *AgentType* as well as the *MessageType* are modeled as direct subtypes of the *BaseObjectType* inheriting the basic properties of an OPC UA *node*. The *MessageType* is further detailed by deriving subtypes of messages such as the *GetOfferMessageType* that represents the negotiation capabilities of an agent in terms of pursuing production steps or the *SetOrderMessageType* for representing concrete production assignments. The process-related type definitions are structured by means of a *domain-specific namespace*. Unlike the *information model namespace*, domain-specific extensions of the OPC UA metamodel focus on the requirements of a certain application domain, e.g. an engineering domain or a special type of process.

As shown in terms of the *MessageType* example, the utilization of the OPC UA metamodel offers a high variety and in-depth modeling capabilities for any kind of objects, entities or even domain ontologies. That being said, the ultimate goal of the OPC UA based representation of multi-agent system becomes clear: The described form of representing software agents in a digital way enables an integration of the agents' capabilities with other services by means continuous semantics and context information.

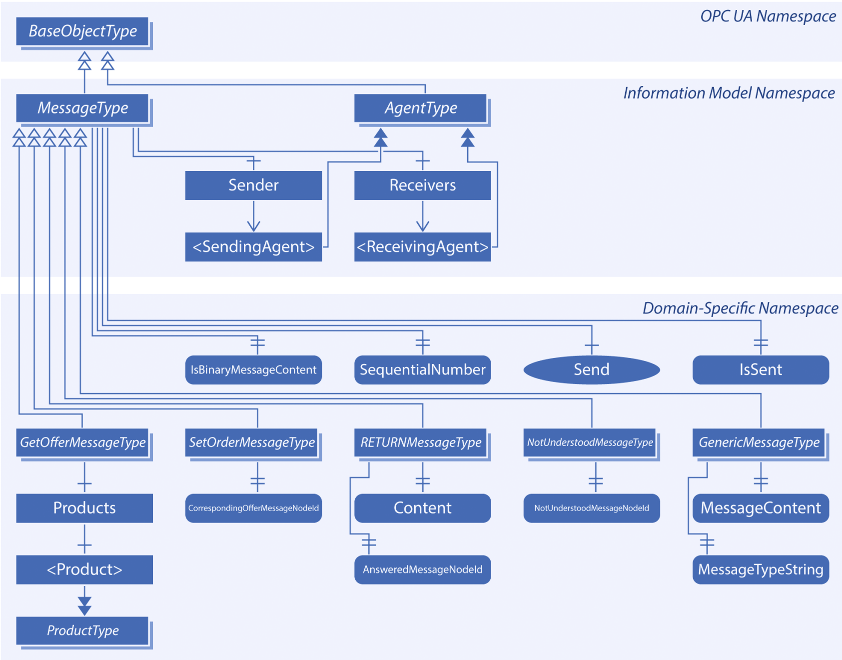

Fig. 4. Detailed *MessageType* specification of the OPC UA agent metamodel

With regard to the manufacturing related context that is in the focus of this work, some specific capabilities of software agents can be pointed out, which can all be represented in accordance to the described modeling concepts:

- The basic functionality of an agent is to sense its direct environment. With regard to a production scenario, this includes the capability to observe the state of surrounding agents, e.g. whether they are occupied, free, etc.
- Another crucial feature of an agent is the capability to receive messages from other entities. The advantage of the presented approach is that a form of generic interoperability can be guaranteed as long as the messages exchanged comply with the metamodel definition of the OPC UA agent model.
- In the same manner as receiving messages, the agents modeled in terms of the presented framework are also capable of placing messages to every OPC UA *node*, e.g. to communicate with other agents or to propagate information to high-level systems of the production planning and organization.
- Furthermore, an agent is capable of processing the content of messages, e.g. to perform actions based on the gained knowledge. The metamodel definition of the message objects ensures that each message is interpretable by the agents and can be processed in accordance to the underlying context, i.e. the specific application domain the agents comply to.
- The capability to process information from the surrounding environment enables agents to take autonomous decisions and accordingly negotiate with other entities. In terms of a manufacturing scenario, this can either be the

negotiation with regard to a production step or the execution of actions such as performing a transport or the conduction of a manufacturing step.

In order to utilize the described capabilities not only in the context of a dedicated subsystem on the shop floor, but also in terms of interactions with other systems of the automation pyramid, the software agents need to be capable of communicating with these higher system. The OPC UA metamodel already provides the basic enablers for this purpose. However, in order to communicate with higher instances, e.g. enterprise resource planning systems, an interface is required that does not only deliver a suitable protocol, but does also provide an understanding of the context information of the agents' application domain.

The purpose of the next section is to describe the design of an interface agent that interconnects agents from MAS in the shop floor to high level ERP systems.

4 OPC UA Based ERP Agents

Traditionally, interface technologies on the shop floor, e.g. enabled by service buses or similar technologies, enable a horizontal integration of the components by providing syntactical and semantic concepts for an information exchange. These concepts are usually focused on proprietary, fixed protocols and communication channels. Even though, these systems provide a functional cooperation of the entities in the field, the information exchange capabilities with higher systems is very often limited to a single interface as seen in Fig. 5.

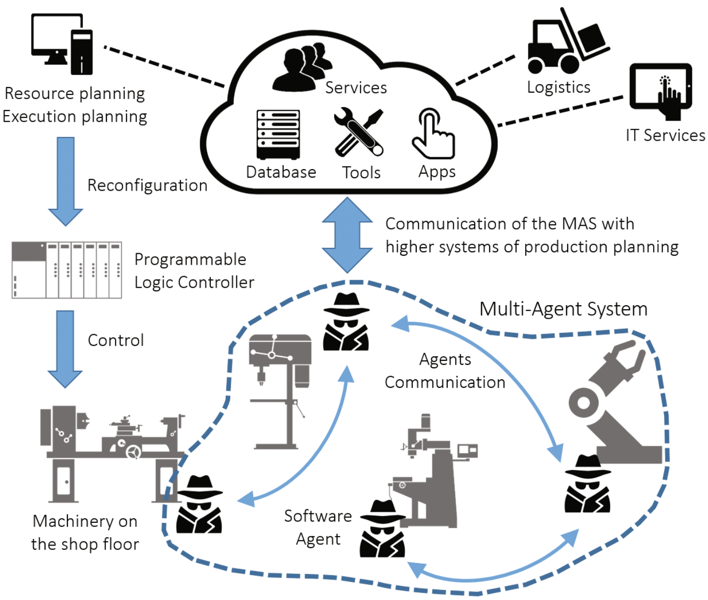

Fig. 5. Interface of tightly coupled MAS with higher systems

The software agents in the field represent certain machines and their capabilities that are shared by means of communicating with each other. However, their connectivity to high-level systems, e.g. Manufacturing Execution System (MES) is depending on a single interface that serves as a fixed gateway between a tightly coupled MAS and all high-level systems. Thus, the information that is changed throughout this interface is generally aggregated and represented in a highly condensed form. Profoundly granular information that could be of major importance during the execution planning, is usually not exchanged.

4.1 ERP System Connectivity Through OPC UA Based Messages

The architecture that results from the semantic information modeling approach that was carried out in terms of this work is depicted in Fig. 6.

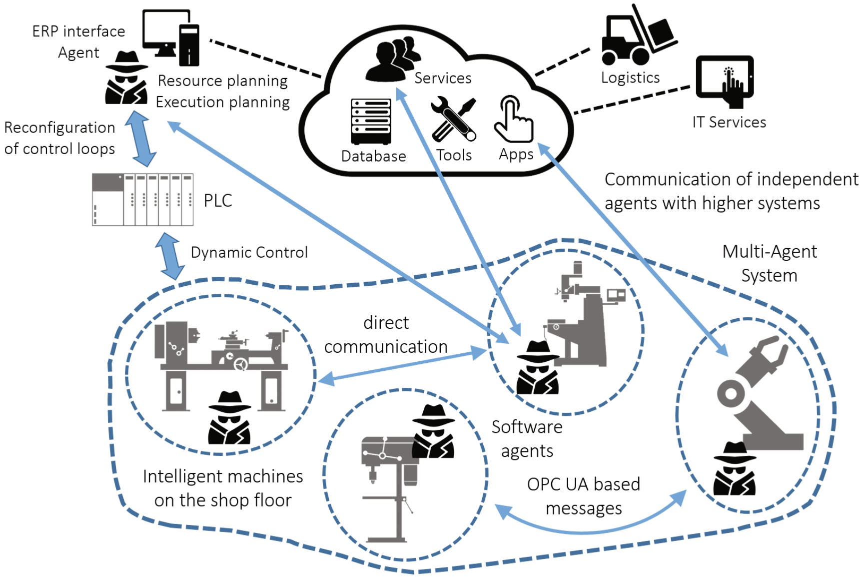

Fig. 6. Generic interoperability of software agents with higher systems

Compared to the MAS architecture in Fig. 5 the resulting architecture shown in Fig. 6 is characterized by higher flexibility. The agents remain their identity, thus still representing manufacturing machines for enabling intelligent utilization of production resources through negotiation and similar cognitive techniques. However, in contrast to the prior architectural approach, rich communication is not limited to an information exchange between agents. Enabled by the usage of a generic message representation based on OPC UA, information exchange can be realized either within the MAS, but also beyond, e.g. by communicating directly with MES or ERP systems. The advantage of this approach is that the actual manufacturing control programs can be adjusted in-process in terms of self-optimization, e.g. for compensating intolerances from earlier production steps through later actions in similar manufacturing sequences.

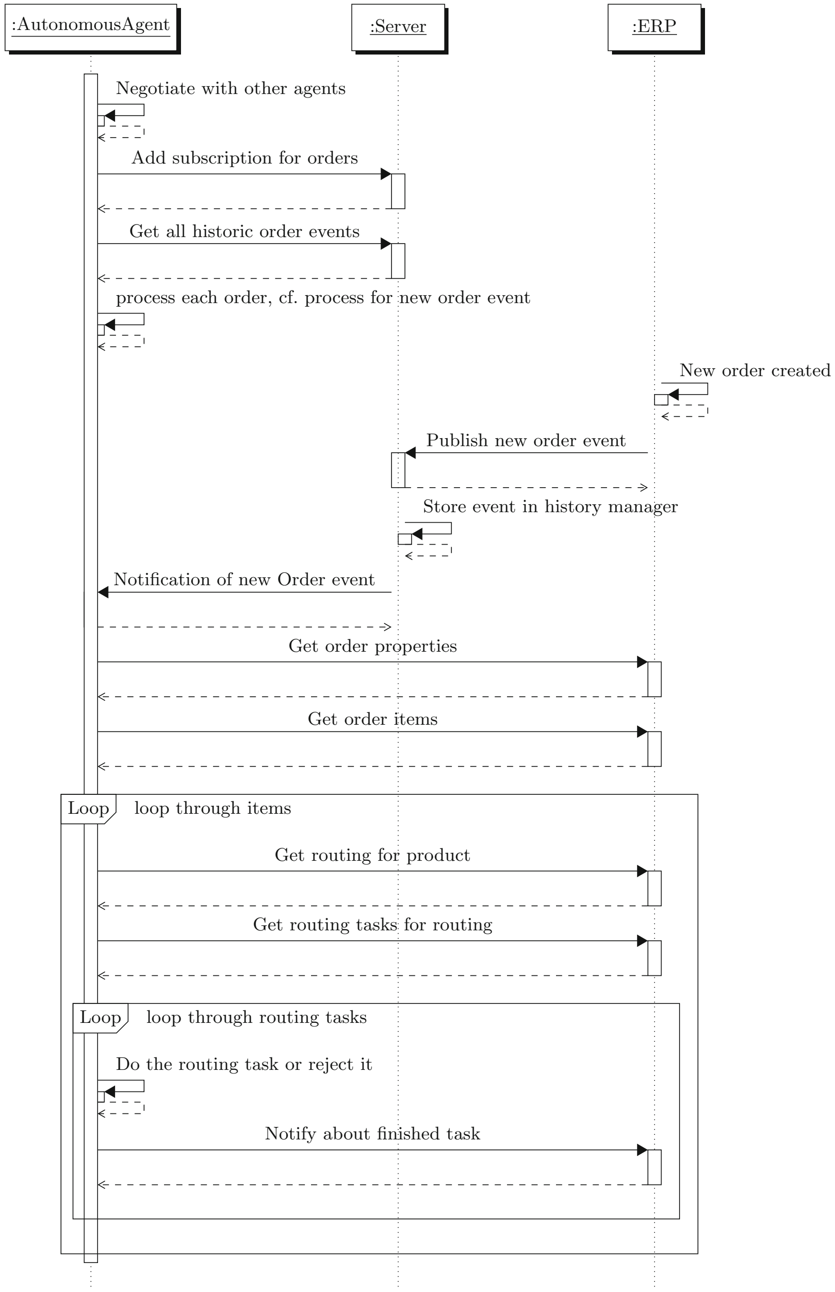

Fig. 7. Sequence diagram of the autonomous agent's process order method

4.2 The ERP Interface Agent

In order to incorporate high-level systems with the MAS in a seamless manner, an interface agent is deployed on the edge of the ERP. This agent provides all services of the underlying ERP system and enables direct interaction with the intelligent software agents in the field. This approach enables an agile planning and execution of the production process as the goals of the manufacturing can be dynamically adjusted with regard to real-time requirements from the ongoing processes. On the other hands, the agents on the shop floor can also benefit from the process knowledge that is integrated within the ERP domain namespace. In this way, the agents are able to focus on the actual process-related optimization strategies without having to deal with structural knowledge, i.e. managing different orders of products or the precise sequence of several production steps. By encapsulating these basic functions from the agents, the entities of the MAS can cooperate based on common production programs and accordingly finish orders more efficiently. Figure 7 shows the logical sequence of an agents cooperating with the ERP system by means of a common OPC UA address space.

In the first step, the agent subscribes to the product orders that are received through the ERP system. If a new agent is logging into an existing MAS the historic order events in the ERP will be returned to make sure that the agent is aware of the current production state. When new orders are placed in the ERP system, the agent is automatically notified about the order by an event.

Based on the order properties, each agent is able to compare the required production steps with their own capabilities and will accordingly decide about a proposition of its availability. The routing of the products that can be obtained for each order specifies the sequence of production steps. When the negotiation process about the execution of production steps between the agents is finished, the manufacturing steps are looped for each product until the product order is completed. Each agent that takes part in the manufacturing notifies the ERP system about the current state of production and about finished production steps. That way the ERP system stays in charge in terms of managing the process flow without infringing the autonomy of agents during the production. Based on this clear distinction of responsibilities a valuable cooperation between managing systems and solutions for embedded intelligence can be realized.

5 Conclusion and Outlook

The high level of technology present in modern industrial production opens up various new applications, many of them based on the usage of data driven technologies. However, there is still a lack of generic integration of these technologies, especially with regard to communication processes between the different parts of a technical system. Most state-of-the-art automation systems are still far away from vertical interoperability through their system architecture.

The current work offers an approach to meet these challenges by presenting an architectural approach that combines two completely different types of technology – communication interface standardization and artificial intelligence.

An incorporation of multi-agent systems and OPC Unified Architecture bears the potential to use adaptive behavior embedded into small devices on the field of a production more effectively by combining the knowledge and capabilities of agents with information from the enterprise layers such as ERP or other high-level systems. Especially the combination with resource planning systems as shown in this work brings decisive advantages in terms of agile planning, reconfiguration and flexible execution scenarios for customized production processes.

The stated solutions are already embedded and evaluated in terms of an Industry 4.0 demonstration test bed presented at the Hannover Fair 2016 in Germany. Future work in the field will focus on the embedding and combination of more sophisticated learning strategies of the agents, which include machine learning solutions, e.g. with regard to predictive maintenance scenarios in combination with an efficient resource planning and product life cycle evaluation.

Acknowledgment. The authors would like to thank the Cluster of Excellence on "Production Technology for High-Wage Countries" at the RWTH Aachen University for the support during the development process of the presented framework.

References

1. Munz, H.: Requirements for time sensitive networks in manufacturing. Why right now? Becuase Industry 4.0 needs it! IEEE 802.1 TSN Standard Meeting (2015). Accessed 22 May 2015
2. Barbosa, J., Leitão, P., Adam, E., Trentesaux, D.: Dynamic self-organization in holonic multi-agent manufacturing systems: The ADACOR evolution. Comput. Ind. **66**, 99–111 (2015)
3. Colombo, A.W., Karnouskos, S.: Towards the factory of the future: a service-oriented cross-layer infrastructure. In: ICT Shaping the World, pp. 65–81. ETSI world class standards, Wiley, Chichester (2009)
4. Feng, S., Stouffer, K., Jurrens, K.: Manufacturing planning and predictive process model integration using software agents. Adv. Eng. Inf. **19**(2), 135–142 (2005)
5. Hannelius, T., Salmenpera, M., Kuikka, S.: Roadmap to adopting OPC UA. In: 6th IEEE International Conference on Industrial Informatics, pp. 756–761 (2008)
6. Hoffmann, M., Thomas, P., Schütz, D., Vogel-Heuser, B., Meisen, T., Jeschke, S.: Semantic integration of multi-agent systems using an OPC UA information modeling approach. In: 14th International Conference on Industrial Informatics (INDIN), Poitiers, France (2016)
7. Jennings, N.R.: On agent-based software engineering. AI **117**(2), 277–296 (2000)
8. Kagermann, H., Wahlster, W., Helbig, J.: Securing the future of German manufacturing industry: recommendations for implementing the strategic initiative INDUSTRIE 4.0 (2013)
9. Lee, J., Lapira, E., Bagheri, B., Kao, H.A.: Recent advances and trends in predictive manufacturing systems in big data environment. Manuf. Lett. **1**(1), 38–41 (2013)
10. Leitão, P., Karnouskos, S., Ribeiro, L., Lee, J., Strasser, T., Colombo, A.W.: Smart agents in industrial cyber-physical systems. Proc. IEEE **104**(5), 1086–1101 (2016)
11. Sauter, T.: The three generations of field-level networks-evolution and compatibility issues. IEEE Trans. Ind. Electron. **57**(11), 3585–3595 (2010)