# Semantic Integration of Multi-Agent Systems using an OPC UA Information Modeling Approach

Max Hoffmann
Philipp Thomas
Institute of Information Management
in Mechanical Engineering
RWTH Aachen University
Dennewartstr. 27, Aachen, Germany
Email: max.hoffmann@ima.rwth-aachen.de

Daniel Schütz
Birgit Vogel-Heuser
Institute of Automation
and Information Systems
Technical University of Munich
Boltzmannstr. 15, Munich, Germany
Email: schuetz@ais.mw.tum.de

Tobias Meisen
Sabina Jeschke
Institute of Information Management
in Mechanical Engineering
RWTH Aachen University
Dennewartstr. 27, Aachen, Germany
Email: sabina.jeschke@ima.rwth-aachen.de

***Abstract*—In terms of current industrial manufacturing sites, a major challenge is to deal with growing complexity by enabling intelligence on the shop floor of existing production processes. A possible solution to reach this goal consists in an integration of smart cyber-physical production systems into the automation systems of a production. One promising approach to do so is based on the agent paradigm. By deploying Multi-Agent Systems into the manufacturing components, each production step is able to gain a self-representation and to achieve intelligent behavior of the entire system. One problem though is the formalization of agent based systems and their communication among each other, which is currently rather hard-coded or application-specific.**

**In this research paper, we propose an architectural approach for a Multi-Agent System that is based on OPC UA as modeling interface and as semantic approach for the integration of agent-based systems into existing manufacturing sites. For this purpose, we define a domain ontology for the representation of intelligent software agents and for the mapping of an agent-based communication by making use of the OPC UA meta model. Due to this conceptual approach, the integration of intelligent entities such as agents into grown manufacturing systems can be performed in a structured and well-defined way as well as by using existing interfaces and semantic standards. The according agent representation intends to upgrade all production resources that can be linked through OPC UA with intelligent behavioral skills. We evaluate the proposed concept by means of an Industry 4.0 demonstrator implementing agents for the representation actual manufacturing machinery based on Raspberry Pi devices.**

## I. INTRODUCTION

Ongoing changes in industrial manufacturing are more and more characterized by calls for an intelligent process design. Thus, one major goal of the digitalization that is currently taking place consists in an upgrade of grown automation environments in terms of Industry 4.0. However, especially in the fields of automation and industrial control, systems are in focus that have been developed throughout decades and which are mostly based on proprietary, tailor-made solutions. These kinds of information and communication infrastructures can only be successfully upgraded to the next level of digitalization if the basic setup and structuring of the enterprise can be maintained while extending the present Information and Communication Technology (ICT) in terms of suitable interfaces and novel information exchange technologies.

Cyber-physical systems intend to bridge this gap between traditional manufacturing organization and intelligent, data-driven approaches by merging the digital world and physical processes. This is realized by embedding intelligent systems into existing machinery or automation equipment. The aim of these smart embedded systems is to collect, integrate and analyze information from various sources of the automation environment to gain a deeper understanding and knowledge about ongoing processes. This knowledge intends to enable real-time optimization of the production processes based on accurate, actual data and their according context information.

A major prerequisite to gather the needed information from manufacturing processes for usage in high-level production organization and planning systems is vertical interoperability between the different layers of the automation pyramid, e.g. between the shop floor and Manufacturing Execution Systems (MES). A key element to reach this horizontal and vertical interoperability is to use one coherent information model as a global context for all data that is collected from the shop floor to the manufacturing control systems.

In this paper, we describe an approach to reach both interoperability between hierarchical layers of a manufacturing environment and intelligent system behavior by embedding intelligent software agents in terms of well-established and widely accepted interface and information modeling standards. For this purpose, we firstly discuss existing applications of software agents for automation systems as well as common semantic interface standards in section II. In section III an approach for semantic integration intelligent software agents by means of an architectural approach and semantic communication is presented. Section IV finally demonstrates the added value of an integrated MAS by means of an industrial use-case.

## II. RELATED WORK

### *A. Software Agents in Industrial Automation*

Multi-Agent Systems have emerged as an integral part in the field of Artificial Intelligence [1]. Inspired by biological phenomena (e. g. an algorithms) MAS are well-known for approaches such as swarm intelligence, in which intelligent entities are able to solve problems in a cooperative way.

Solutions based on MAS have also been adopted to other research fields, especially within engineering sciences. In the domain of manufacturing and automation, intelligent software agents have been applied for supporting decentralized automation systems [2] as well as for increasing the flexibility of production systems and the support of enhanced autonomous functionality [3], for example in terms of coordination tasks [4] and dynamic reconfiguration of processes during operation. Consequently, several research works focus on reconfiguration tasks in case of unforeseen events or system failures, e.g. by enabling adaptive transportation systems [5] [6].

Another concept that focuses on a Plug & Play architecture on the shop floor of manufacturing systems is introduced in [7]. The presented agent-based approach enables a dynamic machine module assignment during operation. The implementation of agents is hereby performed complying with the standards issued by the Foundation of Intelligent Physical Agents (FIPA). The agents controlling the manufacturing plant are on computers that are connected to sensors and actuators of the machine modules. The communication between the software agents in this context is realized using the FIPA Agent Communication Language (ACL). The ACL standard defines the semantics of intelligent software agents talking with each other. Even though this FIPA compliant agent language serves as a standardized and widely accepted communication mechanism, the standard lacks of scalability and flexibility and cannot be easily integrated with existing interface standards. As a consequence, it is the aim of this work to integrate the semantics of agent communication using generic approaches.

### B. *OPC UA enabling intelligent, integrated communication*

A well-established standard to serve interoperability among distributed systems such as automation environments by enabling an overall information modeling approach is the OPC Unified Architecture (OPC UA) interface standard. OPC UA is the successor of the OLE for Process Control (OPC) standard, which had been widely embedded in automation environments for data acquisition purposes throughout the 90's. In contrast to it's predecessor, the OPC UA standard is quipped with integrated information modeling and modern information exchange patterns by means of web services and related methods that follow the concept of service-oriented architectures (SOA). One key feature that sets OPC UA apart from most other communication interfaces is its separation of concerns in terms of data transport and information modeling. OPC UA does not only enable transport of machine or sensor data (e.g. control variables, measurement values and parameters) but also machine-readable and semantic annotation of information [8] that puts production data into an overall context.

Thus, besides information transport capabilities, OPC UA also enables the design and structuring of a full manufacturing environment taking into account all data sources and sinks. By making use of a predefined meta model and address space, all manufacturing resources as well as products or intelligent embedded systems can be assigned to a certain node in a domain-specific and highly scalable namespace.

## III. INTELLIGENT SOFTWARE AGENTS BASED ON OPC UA

The aim of this work is to facilitate a realization of forecasting scenarios inspired by the Industry 4.0 using embedded intelligent entities. By identifying OPC UA as an ideal approach for the integration of semantic production information with distributed systems, the interface standard seems to deliver sufficient capabilities for semantic integration of intelligent software agents with existing production sites.

OPC UA encapsulates information modeling from architectural design. This separation enables an object-oriented modeling of agents and their according communication concepts separately from the modeling of the production environment. Furthermore, the communication paradigm of OPC UA, which is based on client-server relations, facilitates SOA-like peer-to-peer network design that is needed for an efficient agent communication. The OPC UA information modeling approaches are further used to formalize the "language" of agent communication and to enable semantic, context-aware information exchange between the intelligent entities. According to the concepts of OPC UA system design and information modeling approaches, the semantic integration of MAS is performed by firstly defining the architectural approach of agent systems and their according message transport concepts. Secondly, the communication language between agents is specified to enable semantic information exchange within MAS.

### A. *Architecture Approach for MAS based on OPC UA*

The goal of this step is to carry out an architectural design for MAS that enable peer-to-peer communication and at the same time comply with FIPA standards of agent systems. The basic architecture of common MAS is depicted in Fig. 1 [1].

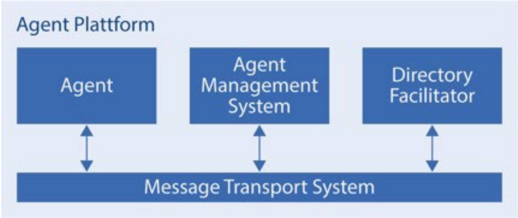


Fig. 1. Basic architecture for the management of an MAS [1]

In the figure, the Agent Management System (AMS) serves as a central instance for the administration of software agents by means of a service registry in SOA. The AMS ensures the registration of all agents and offer white page services that enable discovery of other agents. The Directory Facilitator (DF) manages the abilities of each software agents and intends to facilitate the assignment of tasks using yellow page services.

The communication process between software agents is based on messages. Following the FIPA ACL standard for agent communication, agent messages can be understood and interpreted by an agent that is part of the same MAS as well as by agents of other MAS. To preserve this general comprehensibility of agent messages, the OPC UA based architectural approach for agent communication has been carried out according to the illustration in Fig. 2.

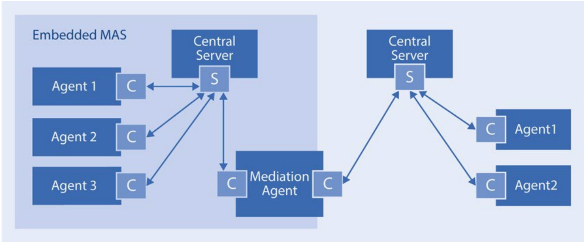


Fig. 2. OPC UA-based architecture of distributed Multi-Agent Systems

In this architecture, the agent are represented by OPC UA clients (”C”) that connect to an OPC UA server (”S”), which consolidates the functionality of the AMS and the DF by managing all agents and their according capabilities. By only routing messages, the OPC UA server preserves the communication autonomy of the agents. The proposed architecture also takes into account communication between different MAS by including a mediation agent that guarantees compliance with the ACL standard while organizing the messages between the central servers of different MAS. A more detailed sequence diagram of the agent communication is shown in Fig. 3.

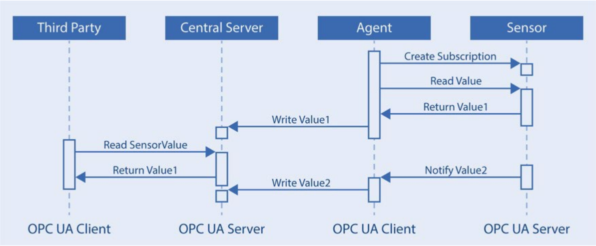


Fig. 3. Message transport between OPC UA agents and servers

In this communication flow diagram, a sensor is represented by an OPC UA server instance which propagates information from the production environment. An OPC UA client that polls this data, also serves as an agent. By making use of the central server, the gathered information can be exchanged between different agents and also be further propagated to third party applications. These third party client applications can be represented by higher information systems, e.g. MES or ERP systems and thus offer an approach for vertical interoperability.

## B. Semantic modeling of messages exchange in MAS

Autonomous and intelligent behavior of agents requires accurate, up-to-date information exchanged by messages. To realize seamless information consistency in MAS, the following modeling steps need to be performed to reach this goal:

1) Development of an agent type node for the representation of an intelligent software agent by means of OPC UA semantics. This is done by an extension of the OPC UA basic meta model (address space).
2) Elaboration of OPC UA-based message type nodes that comply with the FIPA ACL standard.
3) Integration of agent representation nodes with models for the mapping of domain-specific process knowledge.

The initial step is the elaboration of a generic agent based on the OPC UA meta model. The result is shown in Fig. 4.

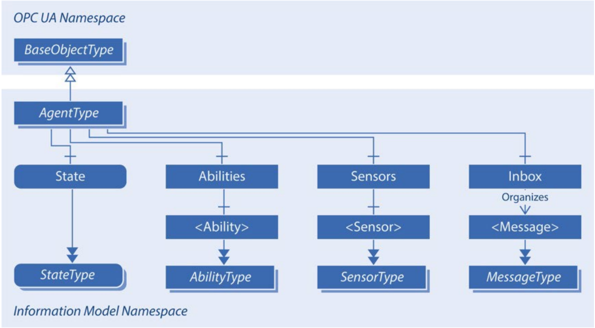
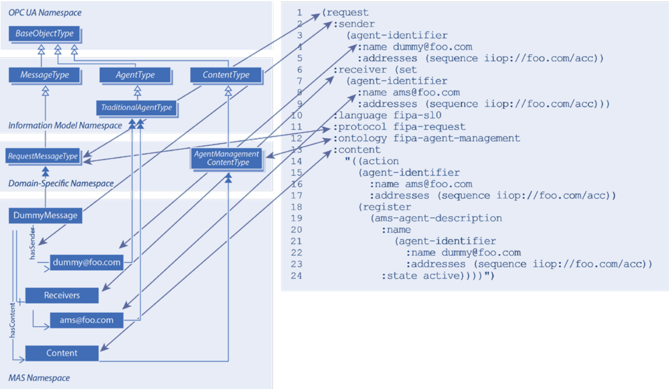


Fig. 4. AgentType specification for the OPC UA agent meta-model

The definition of the AgentType as shown in the model illustration intends to cover all characteristics and capabilities of intelligent software agents. The AgentType is an ObjectType node according to OPC UA meta-model and is derived as a sub type of the BaseObjectType node. That way, the agent type definition can be seamlessly integrated into the address space of an OPC UA application. The agent type definition further contains several HasComponent relations pointing to the sets of *States*, *Abilities*, *Sensors* as well as to the definition of agent *Messages*. Each agent instance is characterized by one state variable, multiple abilities and sensors as well as an inbox for the organization and processing of incoming messages.

In order to complete the model representation of MAS using OPC information UA modeling concept, the agent messaging semantics according to the FIPA ACL standard are modeled in the following. The according information model of the MessageType fulfilling these requirements is depicted in Fig. 5.



```
(request
  :sender
    (agent-identifier
      :name dummy@foo.com
      :addresses (sequence iiop://foo.com/acc))
  :receiver (set
    (agent-identifier
      :name ams@foo.com
      :addresses (sequence iiop://foo.com/acc)))
  :language fipa-sl0
  :protocol fipa-request
  :ontology fipa-agent-management
  :content
    "((action
      (agent-identifier
        :name ams@foo.com
        :addresses (sequence iiop://foo.com/acc))
      (register
        (ams-agent-description
          :name
            (agent-identifier
              :name dummy@foo.com
              :addresses (sequence iiop://foo.com/acc))
          :state active))))")
```

Fig. 5. The ACL compliant MessageType model for OPC UA based messages

The object nodes shown at the left side of Fig. 5 in terms of OPC UA information models spread over multiple namespaces including the MAS namespace and domain-specific content definitions of agent messages, which can be freely extended in terms of domain-specific semantics. Each node complies with the according entity definition in the FIPA ACL standard and points to the according instances shown at the right sight.

## IV. VALIDATION OF AN INTELLIGENT MAS BASED ON OPC UA BY MEANS OF AN INDUSTRIAL USE-CASE

The use-case section of this paper intends to demonstrate the wide area of applications that can be addressed using the presented approach of mapping multi-agent systems and their communication on the OPC UA standard. For this purpose a self-organizing and autonomously reconfigurable process for the production a customized ”yogurt” is examined in the following. After introducing a context-specific information model for the mapping of the yogurt domain, the integration and usage of shop floor information within higher information systems is shown in a further step.

The yogurt production process comprises of multiple production steps, namely *Yogurt Production*, *Yogurt Refinement*, *Cap Engraving* and *Filling* of the product. In this context, the production and refinement need to be performed sequentially, while the cap engraving is done independently in a parallel process. After the refinement and cap engraving are finished, the filling of the yogurt takes place, which represents a process step similar to the assembly in a manufacturing process.

In terms of this work, the yogurt production process, which was initially implemented by [9], has been adapted in the form of a simulated process to assess the usability of OPC UA as modeling and communication paradigm. During the digital manufacturing of the product, all communication is performed by means of OPC UA and based on the derived representation of an MAS. Each production facility is hereby represented by a software agent and instantiated by an OPC UA client.

The simulated process and the according product can be tailored by a customer who is able to specify the properties of the final product by choosing, e.g. the flavor, topping or the cap engraving inscription of the yogurt. This customization process is performed using a web application, which serves as the ERP interface enabling customer-specific information integration into manufacturing system. After the product has been fully specified, the user sends an according production offer into the multi-agent system. The MAS analyzes this offer and derives a production plan based on the needed capabilities in order to fulfill all production steps of the final product. This process is done by means of a coordination agent that routes the messages between the agents representing different production resources. In case of redundant production facilities with the same capabilities, a pricing mechanism randomly picks one of the machines that is capable of the particular production step. After the planning phase is finished, the production starts autonomously and finishes each production step including necessary transportation steps.

The intelligence of our solution consists in an integrated reconfigurability that is characterized by capabilities to reorganize the production in case of unforeseen events. In terms of the particular process that is considered here, the user is able to deactivate a machine during production, i.e. simulate a machinery defect. Following to this, the production process is reconfigured taking into account alternative production resources that are able to compensate the defect machine.

## V. CONCLUSION

In the current work, a representation of Multi-Agent Systems and intelligent software agents has been performed in terms of an OPC UA information modeling approach. The solution provided enables a generic design of intelligent production entities that can be embedded into traditional manufacturing facilities to realize an intelligent behavior and vertical integration by making use of semantic interface standards.

The benefits of this approach result in several advantages for the design of multi-agent systems in automation systems. On the one hand, the proposed concept enables an integrative, generic communication interface between intelligent software agents, on the other hand the software agents gain access to all other production information such as sensor data or key performance indicators of the production process. The agents are accordingly able to integrate current manufacturing data into their intelligent decision-making processes.

Following steps in the research direction of intelligent agents will focus on an enhancement of the agents' autonomy. This is done by enabling concurrent negotiation and competition mechanisms in the MAS as well as learning mechanisms. The agent software demonstrator, which is located at the RWTH Aachen University, will be further advanced by an integration of automation equipment such as PLCs and by an integration of high-level information systems in terms of MES and ERP software modules.

## ACKNOWLEDGMENT

The current work was carried out in terms of the excellence cluster ”Integrative Production Technology for High-Wage Countries” at the RWTH Aachen University and was supported by the German Research Foundation (DFG).

## REFERENCES

[1] M. J. Wooldridge and N. Jennings, *Agent technology: Foundations, applications, and markets*. Berlin and London: Springer, 1998.

[2] W. Lepuschitz, A. Zoitl, M. Valle´e, and M. Merdan, “Toward self-reconfiguration of manufacturing systems using automation agents,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 41, no. 1, pp. 52–69, 2011.

[3] R. W. Brennan, “Toward real-time distributed intelligent control: A survey of research themes and applications,” *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 5, pp. 744–765, 2007.

[4] G. Chen, Z. Yang, and C. P. Low, “Coordinating agents in shop floor environments from a dynamic systems perspective,” *IEEE Transactions on Industrial Informatics*, vol. 2, no. 4, pp. 269–280, 2006.

[5] M. Vallee, M. Merdan, W. Lepuschitz, and G. Koppensteiner, “Decentralized reconfiguration of a flexible transportation system,” *IEEE Transactions on Industrial Informatics*, vol. 7, no. 3, pp. 505–516, 2011.

[6] J. Yan and V. Vyatkin, “Distributed software architecture enabling peer-to-peer communicating controllers,” *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 2200–2209, 2013.

[7] J. Barata, G. Caˆndido, and F. Feijaˇo, “A multiagent based control system applied to an educational shop floor,” in *Information Technology For Balanced Manufacturing Systems*, ser. IFIP Int. Fed. for Information Processing. Springer US, 2006, vol. 220, pp. 119–128.

[8] S.-H. Leitner and W. Mahnke, “Opc ua - service-oriented architecture for industrial applications,” *ABB Corporate Research Center*, 2006.

[9] S. Ulewicz, D. Schu¨tz, and B. Vogel-Heuser, “Design, implementation and evaluation of a hybrid approach for software agents in automation,” in *2012 IEEE 17th Conference on Emerging Technologies & Factory Automation (ETFA 2012)*, 2012, pp. 1–4.