Object-Oriented Communication Model for an Agent-Based Inventory Operations Management

Rafal Cupek, Adam Ziebinski, Lukasz Huczala
Institute of Informatics
Silesian University of Technology
Gliwice, Poland
Rafal.Cupek@polsl.pl

Daniel Grossmann, Markus Bregulla
Ingolstadt University of Applied Sciences
Ingolstadt, Germany
Daniel.Grossmann@thi.de

***Abstract*—this document presents the idea of an autonomous mobile platform that was developed for Inventory Operations Management support that is directly connected and acts as part of a Manufacturing Execution System. The proposed Multi-Agent System supports logistic operations in manufacturing. This article focuses on the object-oriented communication model for a multi-agent logistic system that is based on standardized MES services that are defined according to the ISA95 model and object-oriented communication services implemented in OPC UA standard.**

Keywords-Manufacturing Operations System; MES; Inventory Operations Management; Autonomous Systems; industrial communication; Industry 4.0; OPC UA; ISA95.

I. INTRODUCTION

Inventory Operations Management (IOM) is the process of planning and directing that allows for the easy control and effective flow of materials, semi-products and final products, as well as to the information flow from the source of the information to its destination in order to fulfil a customer's requirements. One of the main goals of an IOM is to minimize the final cost of products by reducing the costs of transportation, management and storage in warehouses [1].

IOM execution is supported by logistics systems that help to improve the processes for the management of an enterprise and to perform the necessary analyses that are necessary to reduce costs. They include analyses of the loading, transferring, unloading and storage of products in the supply cycles for both the production and delivery stages. Logistics systems can be grouped according to different criteria [2] such as: functionalities (systems of supply, production or distribution), structural-decision-functional criteria (systems of planning, controlling and organizing) or hierarchical criteria (normative, strategic and operating systems). As was shown in [3], autonomous vehicles are one of key components in the development of flexible and efficient transport systems for logistics and industrial site management applications. Such a system needs the distribution of intelligence across devices as well as logistics and business applications.

An IOM cannot function in isolation from a Manufacturing Execution System (MES). One of the commonly accepted definitions of MES activities can be found in the set of documents that is managed by MESA, the International Manufacturing Enterprise Solutions Association, and expressed as the ANSI/ISA95 (IEC/ISO 62264) norms that are the international standard for the integration of enterprise and control systems [4].

ISA95 defines the MES data structure and MES services that are related to manufacturing operations: defining the product, forecasting production, managing production capability and evaluating production performances. ISA95 consists of models and terminology and describes the information that is exchanged between the systems for sales, finance and logistics and the systems for production, maintenance and quality. This information is structured in the form of UML (Unified Modelling Language) models, which are the basis for the development of standard interfaces between ERP (Enterprise Resource Planning) and MES systems. ISA95 is built on an object-oriented model that defines the interface between the control systems and a business application. It also defines the services that are required for the manufacturing support that is designed according to the object-oriented model [5].

One of the leading examples of the Service-Oriented communication standards that are used in industry is OPC UA (Open Production Connectivity Unified Architecture). OPC UA is a service-based architecture that relies on Web Services for communication with enterprise management systems and TCP-based communication for communication with control and HMI (Human Machine Interfaces) [6, 7]. OPC is maintained by the OPC Foundation [8] and is recommended by Industry 4.0 guidelines. As was shown in [9], OPC UA can be applied as a communication interface between an MES (Manufacturing Execution System) and the real-time devices that are used for logistics operations. OPC UA is based on an object-oriented model that allows a flexible communication interface to be created that can be used in heterogeneous logistics systems. OPC UA supports logistics data modelling and annotating raw input data with useful semantic information that supports logistics decisions.

This article focuses on the object-oriented communication interface model for an intelligent agent system that is designed for logistic operations. The proposed model follows the existing manufacturing standards and reflects the system requirements that have been defined for cyber-physical systems that are developed in accordance with the model defined by Industry 4.0 [10]. The authors propose an agent-based architecture that is connected to an MES according to the ISA 95 standard. The proposed communication interface is based on OPC UA with a special focus on the ISA95 OPC UA information model that is accepted in the guidelines of Industry 4.0.

The rest of this paper is organized as follow: the agent-based architecture concept of the Autonomous Mobile Platform (AMP) is presented in chapter two. Some of the details of the implementation of the AMP platform are given in chapter three. On the one hand, an AMP is an autonomous device that is designed to support material transportation, while on the other hand, an AMP is part of the inventory operations management support and from this point of view, it is part of a distributed agent-based MES. The concept of a data model that is based on the OPC UA communication interface and joins the AMP and MES according to the ISA95 architecture is described in chapter four. The conclusions are presented in chapter five.

II. AGENT-BASED ARCHITECTURE FOR INVENTORY OPERATIONS MANAGEMENT (IOM)

In order to implement our Distributed Agent-Based Architecture concept, we decided to use the inventory operations management activity model that was proposed in the third part of the ISA95 standard. This model reflects the four main information streams that are exchanged between an MES and enterprise management systems. Inventory definitions describe the rules and information that are associated with the movement and storage of materials. These rules may be location specific, equipment specific, physical asset specific or material specific. Inventory capability is a capability measure of the ability to handle materials for specific time horizons and is characterized by the type of material, available storage space (or volume) and type of storage. An inventory request is used to define requests to transfer materials. An inventory response is used to respond to an inventory request and indicates the completion status (successful or unsuccessful) of the request [11].

Although the structure of the activity that is proposed in the ISA95 seems fine for individual functions, it cannot be effectively implemented in multi-agent systems. ISA95 defines detailed inventory scheduling as a collection of activities that take inventory requests and generate work schedules for inventory. This reflects the hierarchical approach top to bottom that is used in planning systems but does not reflect the dynamics of a real logistics system. Instead of duplicating the structure that is proposed by the ISA95, the authors propose the division of the inventory operations management between agents as shown in Fig. 1.

The Logistics Agent (LA) is a communication entry point that is responsible for collecting inventory requests from the MES and is the owner of the Transportation Request list. Detailed inventory scheduling is created by Delivery Agents (DA) and is achieved by distributed and collaborative work. Each inventory request generates one instance of a Delivery Agent (DA) that is responsible for the realization of the delivery. Delivery Agents fulfil the detailed inventory scheduling and Inventory dispatching functionality as defined by ISA95. They analyze transportation possibilities and the costs that are offered by Transportation Agents (TA). Transportation Agents are holons that are directly coupled with mobile platforms. Each mobile platform has its own TA, which manages the Confirmed Orders list and the Required Orders list. Each delivery request that is sent by a DA is placed on the Required Orders list. Deliveries that are confirmed by a DA are moved to the Confirmed Orders list. A DA can also cancel a delivery order that has been accepted by a TA for realization but has not been started yet. In such a case, the delivery record from the Confirmed Orders list is moved back to the Required Orders list.

A DA bases its decisions on the delivery costs that are exposed by a TA. A DA sends a response to the TA that can be either a Confirmation or a Cancellation of the delivery service that was proposed by the TA. Every new Order that is accepted by the TA can change the cost for other Confirmed Orders. This information is sent back from the TA to interested DAs including both owners of Confirmed Orders and Required Orders. Based on these, the DA can decide to change the TA and send the message to cancel the delivery. This may result in a cost change for other orders. To make the system more stable, the Cancellation cost, which is a function of the Time to start transportation service, has to be added to the total delivery cost.

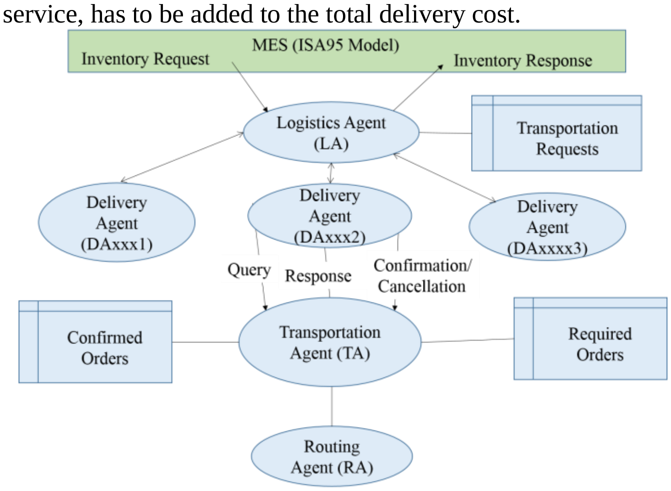

Figure 1. Proposed IT architecture for an agent-based IOM system.

III. ARCHITECTURE FOR AN AUTONOMUS MOBILE PLATFORM (AMP)

An AMP is designed as a modular project that is based on a simple mobile platform, an engine control module and a movement control system. The authors assume that the mobile platform and engine controls are one of the commercially available solutions and the control system is realized on the Raspberry Pi system.

The AMP is supervised and monitored by the Distributed Logistics System of a Transportation Agent (TA). In the proposed solution, the TA operates as a holon for the AMP and is responsible for the execution of the transportation services that are ordered by the DAs. Motion control of the AMP is executed through an application running on the

RaspberryPI that interprets the signals that are collected from the sensors that are installed on the platform and they perform the communication tasks that are received from the TA. Communication between the logistics system and the TA based on the RaspberryPI platform is performed via a Wi-Fi/GPRS. The TA is able export its services to the logistics system.

The movement control system of an AMP allows it to drive a vehicle forward and backward, turn left and right and rotate in place. In order to achieve this functionality, an AMP can be equipped (depending on the AMP configuration) with a set of sensors that includes a Gyroscope, Accelerometer, GPS, Encoders, Short Range Radar and Lidar, Ultrasound and Camera. It was assumed that the AMP will be moved along a known route on a flat terrain – pavement or a corridor in a building. The Transportation Agent allows data to be exchanged between the AMP and other agents. A Routing Agent (RA) is responsible for preparing the route for the Transportation Agent. The RA will send part of the route parameters to the TA based on the receipt of subsequent jobs – short segments of the route include movement direction, velocity, distance, order of delivery, etc. Next the Transportation Agent sends this information to the AMP controller.

Another functionality is the continuous monitoring of the AMP route by Transportation Agent using the RA. This will allow information from the sensors, camera and the control signals to be written into the database. That data will allow the Transportation Agent to monitor and analyze the route that is travelled, The RaspberryPI application, which supports real time functions, is responsible for the movement, position, velocity, and detection of the distance to obstacles and current consumption.

An additional functionality is the continuous recording of the route and sending this information to the Transportation Agent. The information that is sent includes data from the camera, sensors, orders that were issued, power that was consumed, etc. When the system on the AMP detects an obstacle, it will try to bypass the current obstacle, and in the event of failure, the AMP will scan the surrounding area and try to find another way to drive to the designated point. In the event of failure to find good way in this mode, the Transportation Agent will ask the RA to prepare a new route. The system will go into manual mode if the RA cannot prepare a correct route.

IV. Open Production Connectivity Unified Architecture (OPC UA) Interface for Inventory Operations Management

OPC UA is an object-oriented and service-based communication interface. OPC UA joins the functionality that was offered by the previous sets of OPC specifications that are now integrated into a unified service set that is defined on an abstract level. The OPC UA services are organized into ten Service Sets: Discovery, SecureChannel, Session, NodeManagement, View, Query, Attribute, Method, MonitoredItem and Subscription. The service sets contain 37 services of which 16 are used for actual information exchange and 21 are used to manage the communication infrastructure. All of the OPC UA functionality is based on these 37 services, which represent all of the possible interactions between a UA client and UA server applications.

The services that are defined by OPC UA are platform independent and are defined on an abstract level. There are various OPC stack implementations including binary UA TCP encoding and an XML encoding with the SOAP/ HTTP transport protocol. Information exchange is managed by a Secure Channel, which defines the long-running logical connection between an OPC UA Client and Server. This channel maintains a set of Public-key infrastructure (PKI) keys that are known only to the Client and the Server and that are used to authenticate and encrypt messages that are sent across the network.

Information exchange is started by establishing a secure connection between the Client and the Server so that a Client can then browse or query a server’s address space if necessary. Client-Server Sessions are defined over underlying transport layers and may be managed by a Session Service Set. OPC UA communication uses the Session Service, while the OPC UA defines the session level services – Create Session, Activate Session and Close Session services. Such a structure supports the interfaces robustly in the event of communication errors because sessions can survive communication channel breaks. The services of OPC UA are defined on an abstract level. They have already been implemented with a binary UA TCP encoding and an XML encoding with the SOAP/ HTTP transport protocol. Information exchange is managed by a Secure Channel, which defines the long-running logical connection between an OPC UA Client and Server. This channel maintains a set of Public-key infrastructure (PKI) keys that are only known to the Client and the Server and that are used to authenticate and encrypt the messages that are sent across the network.

Although Read/Write services, which are defined in an Attribute Service Set, allow Clients to have direct information access, a more efficient communication may be established based on the well-known Publisher/Subscriber model that is supported by the MonitoredItem and Subscription service sets. In this model, a Client can subscribe to Server parameters for a given set of MonitoredItems – Sampling Interval (ms), Queue Size and Filter Conditions that define the Trigger condition (status, value/status, source timestamp/value/status) and Deadband (Absolute, Percent). According to the tracking conditions that are defined, an OPC UA Server Client send Publish requests (there is no longer call back mechanism) for given Session to the server. Data transfers are optimized by grouping modified MonitoredItems within sessions for efficiency. Published request responses are not sent immediately, but are queried by the Server, whereas a Publish Response is sent back according to the Subscription’s publishing interval. NotificationMessages contain Notifications of any monitored items that have not yet been reported to the Client. If no notification is available, the server sends a life-ping to the client.

OPC UA is scalable and can be used with servers that have high computational power and rich resources but it can also be applied in smart sensors with very limited resources. The individual features are grouped into ConformanceUnits, which are further grouped into Profiles. This allows the actual service set to be adjusted to the application requirements on the one hand and to the hardware limitations on the other. One example of profiles with very low hardware requirements is the OPC UA Nano Embedded Device profile, which describes a reduced functionality and simplified set of services. This profile is fully compliant with the other parts of the OPC UA standard and can be run on platforms with very limited resources. In [12], it was demonstrated that such an environment can even be reduced to a simple FPGA platform that has very limited hardware and software resources.

The application profiles are defined based on the basic service sets. They reflect the main areas of the application of OPC UA – Data Access, which gives subscription-based information about the current values of the parameters that are required by the OPC UA Client; Alarms & Conditions, which are organized as event-based communications that are restricted to the area of interests that is defined by the client; a Historical Access interface for and Programs interface, which allows the server-side application to be controlled by commands that are sent by the client. OPC UA has been an IEC standard (IEC 62541) since 2012.

Nowadays, the OPC Foundation supports different working groups that are involved in the development of the industry-leading specifications, technologies, certifications and processes that are related to OPC UA. One of them is the ISA95 Working Group, which is responsible for defining and maintaining the ISA95 OPC UA information model. The first release of the ISA95 OPC UA specifications includes support for the following ISA95 [4] models: Physical Assets, Equipment, Personnel and Material Handling. This mapping of the abstract ISA95 model provides a high-speed, secure information flow from the lowest levels of the automation hierarchy to the Manufacturing Execution Systems (MES) and Enterprise Resource Planning (ERP) systems .

The features above allow a scalable OPC UA address space that can be implemented both a mobile platform with limited resources and on MES servers that have far more hardware and software resources to be designed. The proposed OPC UA-based communication model consists of a set of OPC UA servers. Each OPC UA server is directly connected with one agent: LA, DA, TA and RA. Each of the agents also has an OPC UA client that is used for communication with the OPC UA servers as shown in Fig. 2.

The OPC UA address space is, by definition, distributed and can include many servers and many clients that are connected by references (if part of information is stored on another server) and subscriptions (to define the information that is important to a given client). Three powerful mechanisms that are offered by OPC UA were used for this purpose:

An OPC UA-type definition offers the possibility to create the object-oriented data structures that their clients need with the possibility to discover the object data structures that are managed by servers. The OPC UA address space includes knowledge about the organization of the available information and presents this knowledge to any application that is connected on the client side. A client can use object-oriented data types to discover their definitions, which are stored on the server. The references and cross-server references are part of the type definition. Types organize data into a structure that includes object-oriented mechanisms such as subtyping and type inheritance. Types can be used for objects, variables, data and reference definitions.

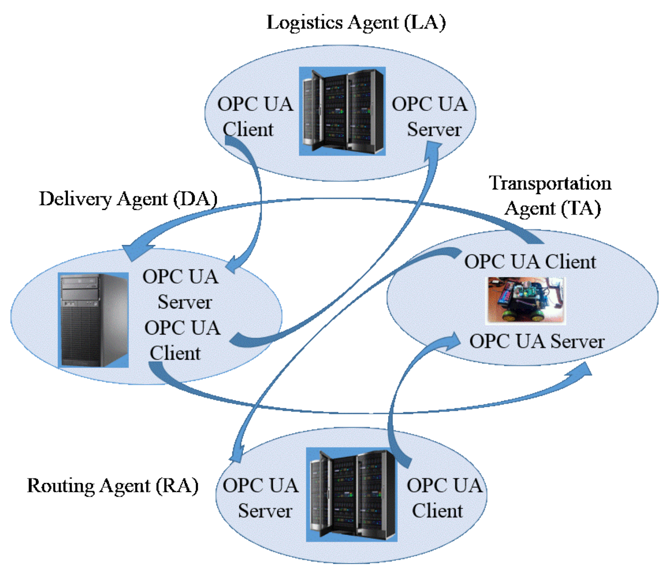

Figure 2. OPC UA-based architecture for a multi-agent IOM system.

The AMP address space defines its own types, which are related to its IOM functionality, such as details about materials, production orders, production schedule and other information that is related to logistics operations. These types are defined according to the functionality of the AMP, the ISA 95 model and the IOM information documents that are exchanged with the Enterprise Management System (ERP), which are based on SAP software (as presented in an example below). The OPC UA address space part that is defined for an AMP reflects the information that is related to route planning such as road maps, obstacle identification and information about other vehicles, etc. Actual information is presented by objects that reflect the operations that are carried by the IOM on the one hand and from the objects that describe the current situation that at the place of operation of AMP on the other hand.

Type definitions reflect the real situation that is presented by the objects that are class instances according to given types. Objects reflect actual information about variables, properties and methods. Variables are used to present the real process signals that change during the execution of the process, the properties that are used to describe an object and in a case in which more complicated activity on the object is needed, the methods that are used. The object-oriented structure is also supported by events mechanisms, which means that there is no longer a need for cyclical data pulling and continuous object state checking. The information is automatically sent to the client that has subscribed to a given event and that receives the required message in the event of its evidence. The subscription mechanism is used to track all kinds of data changes in a consistent way. These include Variables Values, Aggregated Values and Events. According to the SOA model, subscriptions can be managed by the OPC UA MonitoredItem Service Set.

The OPC UA address space for the whole MES can be very complicated. The physical memory of an AMP and communication bandwidth makes it impossible to access all of the IOM information in every TA. This problem can be solved by using a reference-based view mechanism. The view mechanism, which is supported by OPC UA, allows the whole address space that is managed by the OPC UA servers to be reduced to only the information that is necessary for a given client. Views help to organize large data structures in order to present the information that is important in given contexts. In the case of an AMP, it can be used to limit the presentation of information to only the scope that is necessary for a given mobile platform. The information about the IOM is limited to the area that is in the domain of a given TA including any possible transport services. The selected part of OPC UA address space that presents the type definition of the TA and AMP objects is shown in Fig. 3.

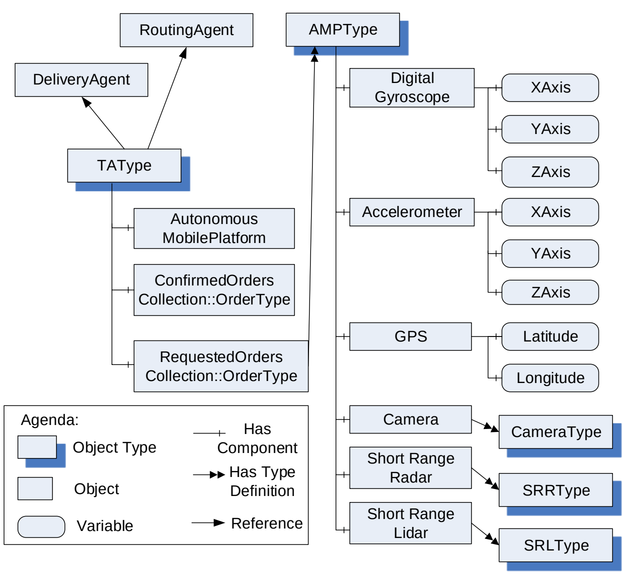

Figure 3. OPC UA address space for an AMP (selected view).

Each TA creates one instance of TAType that is hosted by the OPC UA server. The Information that is presented is used by the DA (collections of Orders) as well as by the AMP in order to exchange information about the ordered transport services and is used for optimal delivery routing. In addition, the TAType contains the AMPType object that presents information related to the location and movement of the mobile platform. This object is also hosted by an OPCUA Server that is owned by the TA. This information is required for the TA to calculate delivery cost.

The productive implementation of the system requires data exchange with the SAP Warehouse Management system, and therefore the SAP standard message types had to be created in the OPC UA address space. The SAP transfer order contains all of the information that is required in order to execute the physical transfer of materials into the warehouse, out of the warehouse or from one storage bin to another storage bin within the warehouse. When you confirm a transfer order, you inform the system that it has been processed and that the goods have arrived at the intended destination.

A transfer order contains all of the necessary information on a planned movement of goods. Additionally, a material definition consisting of the size, weight, transport requirements, etc. is accessible via separate message channel. The SAP data is mapped into an ISA95 MaterialClassType. The required messages in IDOC standard are listed in Table 1, which is based on SAP requirements.

TABLE I. INFORMATION MODEL FOR TRANSFER ORDER IN OPC UA

SAP IDoc message		**IDoc data segment**	
Transfer orders	WMTORD	E2LTORH E2LTORI	Transfer order header Transfer order items
Confirmation of transfer orders	WMTOCO	E2LTCOX E2LTCOH E2LTCOI	Confirm entire storage unit (multiple orders) Confirm entire transfer order Confirmation of a transfer order item
Cancellation request	WMCATO	E2LTCAH E2LTCAI	Header data Item data
Material master	MATMAS	E1MARCM E1MARDM E1MARMM E1MEANM	Material Class segment Warehouse / batch segment Units of measure European Article Number

There are two possible scenarios for creating a message transfer between the Logistics Agent and the Warehouse Management system: OPC UA connection by means of SAP Plant Connectivity (PCo) – IDOC transformation on the SAP side and EDI connection by means of, e. g., IBM InfoSphere Information Server – IDOC transformation on LA side.

The OPC UA data model that is selected has to reflect both the ISA95 and SAP transfer order business process and the data types of the management system. The information model that is proposed for a Transfer Order in the OPC UA address space is presented Fig. 4.

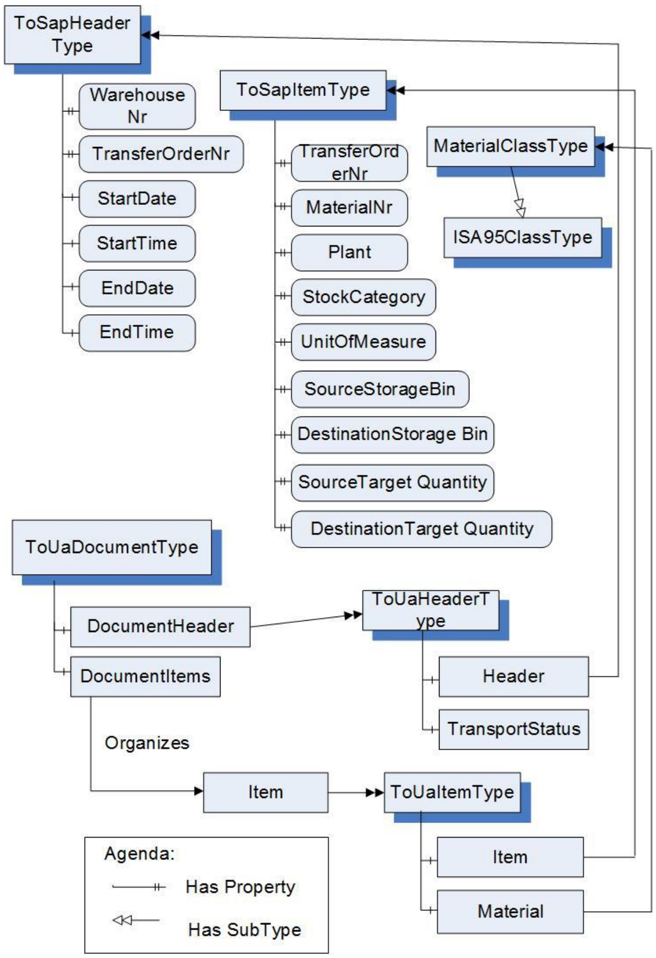

Figure 4. Information model for a Transfer Order in OPC UA [23].

The operations request model for ISA 95 [13] defines the following attributes for operations request objects:

- ID – A unique identification for the operations request
- Description – additional information and descriptions of the operations request
- Operations type – in our case Inventory
- Start Time – When the operation is to be started
- End Time – When the operation is to be completed
- Priority – The priority of the request
- Hierarchy Scope – Identifies where the exchanged information fits within the role-based equipment hierarchy
- Operations Definition ID – Identifies the associated Operations definition that are to be used
- Request State – Indicates the state of the operations request.

V. CONCLUSIONS

The proposed model for IOM systems is based on a multi-agent architecture that is controlled by the flow of events. It is based on the hierarchical model that was proposed by ISA95. For the ERP (SAP) information is presented in a model that is compatible with an ISA95 Operations request. Detailed Inventory scheduling and Inventory dispatching are created and modified dynamically via the interaction of Delivery Agents and Transportation Agents. A detailed communication model is proposed based on the distributed OPC UA address space definition that was adjusted for a multi-agent environment. The next planned step is to validate the stability and efficiency of the model. The authors plan to reach this goal by performing a simulation.

ACKNOWLEDGMENT

This work was supported by the European Union from the FP7-PEOPLE-2013-IAPP AutoUniMo project "Automotive Production Engineering Unified Perspective based on Data Mining Methods and Virtual Factory Model" (grant agreement no: 612207).

REFERENCES

[1] C. F. Daganzo, Logistics Systems Anaysis, Springer-Verlag, 1999.
[2] T. Gudehus, H. Kotzab, Comprehensive Logistics, Springer-Verlag, 2009.
[3] Andreasson, Henrik, et al. "Autonomous Transport Vehicles: Where We Are and What Is Missing. " Robotics & Automation Magazine, IEEE 22. 1 (2015): 64-75.
[4] http://www. isa-95.com.
[5] B.Scholten, The road to integration: A guide to applying the ISA-95 standard in manufacturing. Isa, 2007.
[6] W. Mahnke, S. -H. Leitner and M. Damm, OPC Unified Architecture, Springer-Verlag Berlin Heildelberg New York, 2009.
[7] J. Lange, F. Iwanitz, T. J. Burke, "OPC – From Data Access to Unified Architecture", VDE Verlag, 2010.
[8] https://opcfoundation.org
[9] A.Maka, R.Cupek, and J.Rosner, "OPC UA Object-Oriented Model for Public Transportation System. " Computer Modeling and Simulation (EMS), 2011 Fifth UKSim European Symposium on. IEEE, 2011.
[10] N.Jazdi, "Cyber physical systems in the context of Industry 4. 0. " Automation, Quality and Testing, Robotics, 2014 IEEE International Conference on. IEEE, 2014.
[11] ANSI/ISA95. 00. 03-2013 (IEC 62264-3 Modified) Enterprise-Control System Integration − Part 3: Activity Models of Manufacturing Operations Management; Approved 8 July 2013.
[12] R. Cupek, A. Ziebinski, and M. Franek, "FPGA BASED OPC UA EMBEDDED INDUSTRIAL DATA SERVER IMPLEMENTATION. " Journal of Circuits, Systems and Computers 22. 08 (2013).
[13] ANSI/ISA95. 00. 02-2010 (IEC 62264-2 Mod) Enterprise-Control System Integration − Part 2: Object Model Attributes; Approved 13 May 2010.