UDC 004.771

Thorsten Schöler, Prof. Dr.-Ing.,
Lucas Kögel, Sebastian Pröll

CPS/DATA SCIENCE APPROACH FOR SMART FACTORY AND BUILDING AUTOMATION

Abstract. *Generic hardware/software system architecture for professional Internet of Things applications has been designed and implemented. The system architecture is based on a multi-agent system approach and uses event-based data stream management technologies for data fusion. The described approach has been validated in two real-world settings: A cyber-physical production systems (CPPS) and a smart building closed-loop control system with simulation support (CyPhREE). Acquired event-based data is integrated and used to control and optimise system processes using data science approaches. Finally, technologies and concepts from data science, big data analysis and data mining of the gathered data are described and their integration is proposed.*

Keywords: *Cyber-Physical System, Software Agent System, Production System, Building Control, Event-based, CEP, Data Science, Big Data*

Торстен Шёлер, проф. др.-инж.,
Лукас Кёгель, Себастиан Прёлль

ПОДХОД НА ОСНОВЕ КИБЕР-ФИЗИЧЕСКИХ СИСТЕМ И НАУКИ О ДАННЫХ ДЛЯ АВТОМАТИЗАЦИИ УМНЫХ ЗАВОДОВ И ЗДАНИЙ

Аннотация. *Спроектирована и разработана общая аппаратная и программная архитектура системы для профессиональных приложений Интернета вещей. Архитектура системы основана на подходе мультиагентных систем и использует технологию основанного на событиях потока данных для слияния данных. Описанный подход был проверен для двух практических приложений: кибер-физической производственной системы и системы управления умным зданием с обратной связью и поддержкой симуляции. Полученные, основанные на событиях, данные интегрированы и использованы для управления и оптимизации системных процессов, используя подходы на основе науки о данных. Наконец, описаны технологии и концепты из науки о данных, анализа больших данных и data mining собранных данных и предложена их интеграция.*

Ключевые слова: *кибер-физическая система, системы программных агентов, производственные системы, управление зданиями, события, CEP, наука о данных, большие данные*

Торстен Шолер, проф. др.-інж.,
Лукас Кегель, Себастіан Прьолль

ПІДХІД НА ОСНОВІ КІБЕРФІЗИЧНИХ СИСТЕМ І НАУКИ ПРО ДАНІ ДЛЯ АВТОМАТИЗАЦІЇ РОЗУМНИХ ЗАВОДІВ І БУДІВЕЛЬ

Анотація. *Спроектована і розроблена загальна апаратна і програмна архітектура системи для професійних застосовань Інтернету речей. Архітектура системи заснована на підході мультиагентних систем і використовує технологію заснованого на подіях потоку даних для злиття даних. Описаний підхід був перевірений для двох практичних застосовань: кібер-фізичної виробничої системи і системи управління розумним будинком зі зворотним зв'язком і підтримкою симуляції. Отримані засновані на подіях даних інтегровані і використані для управління і оптимізації системних процесів використовуючи підходи на основі науки про дані. Нарешті, описані технології та концепти з науки про дані, аналізу великих даних і data mining зібраних даних і запропонована їх інтеграція.*

Ключові слова: *кібер-фізична система, системи програмних агентів, виробничі системи, управління будівлями, події, CEP, наука про дані, великі дані*

INTRODUCTION

The Internet of Things (IoT) brings objects from the physical world together with their counterparts in the cyber world. Complex, autonomous and intelligent distributed systems are no longer just monitoring physical objects or processes. More and more of these systems are controlling objects and processes. These complex, autonomous systems are called cyber-physical systems (CPS) (Schöler et al., 2013). Among the technologies used to implement CPS are machine-to-machine communication technologies, such as MQTT (Message Queue Telemetry Transport), distributed software architectures (like software agent systems), and data science approaches for managing vast amounts of event-based data (Complex Event Processing, Big Data and Deep Learning).

Our approach combines those technologies into versatile software architecture for IoT applications. In the following, our system approach will be introduced and validated in two selected use-cases.

AGENT-BASED SYSTEM ARCHITECTURE

Based on the concepts of SEPIA (Schöler, 2014), the heart of our system is a multi-agent system (Jadex[1]) which ties together all necessary functionalities and services. It provides e. g. agent-based services for data integration, data analytics and visualisation. Agents and other subsystems communicate either via agent-based protocols or via MQTT.

Fig. 1 show how our approach is used to implement a cyber-physical production system (CPPS).

The myJoghurt CPPS demonstrator system is a test-bed for distributed manufacturing of highly customised yoghurt products (Vogel-Heuser et al., 2015). Additionally to the already described technologies, our CPPS uses an OWL DL[2] knowledge base as semantic world model as well as an agent-based coordination mechanism (i. e. Contract Net).

As one of the production locations in Fig. 1, our system is able to accept orders to manufacture parts of yoghurt products or whole product orders. The orders, addressed to the production location in Augsburg, are fetched from the myJoghurt application, parsed into a standard JSON format and forwarded into the production system via MQTT.

In order to coordinate multiple orders, a dedicated Contact Agent is implemented. When a new order is detected on a specified MQTT channel the software agent system forwards the orders to a representative Order Agent, which is responsible to initiate all working processes needed for this particular order, including the initialisation of a new Workpiece with a Workpiece Agent as a representative. Each Workpiece Agent has knowledge about all process steps which have to be processed to finish this Workpiece. In order to do that autonomously each Workpiece Agent communicates with the Machine Agents via the Contract Net protocol mechanism. For each step which has to be processed, the Workpiece Agent calls for proposals from the Machine Agents. Each Machine Agent represents a specific part of the industry lane and is capable of processing different types of steps, which may can alternate after a short changeover time. According to e. g. changeover times and other metrics the Workpiece Agent is able to evaluate the proposals and returns an acknowledgement to the best fitting one.

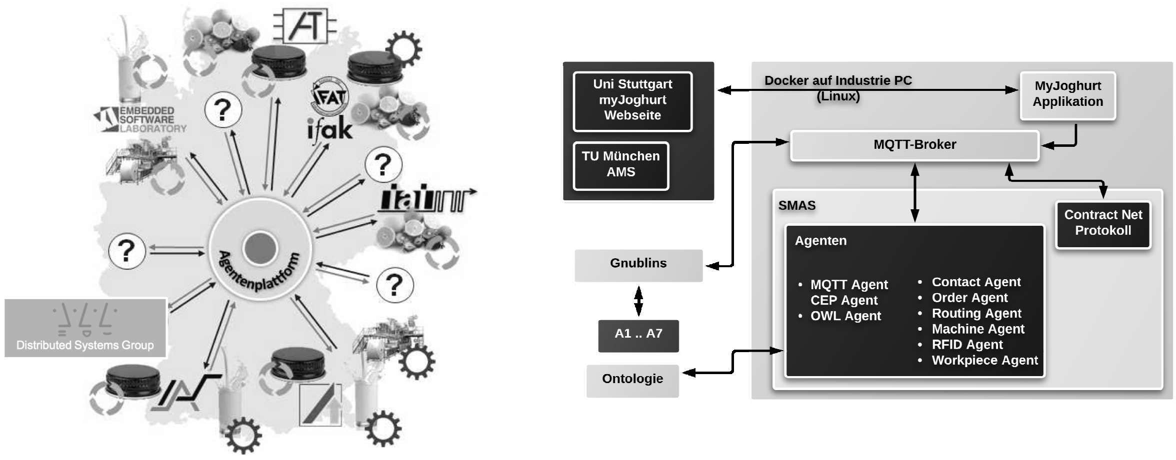

Fig. 1. System architecture my Joghurt CPPS demonstrator

[1] *More about Jadex can be found under https://www.activecomponents.org/ (Last accessed 2016-04-14).*

[2] *Web Ontology Language Description Logic*

To locate and transport each workpiece to its righteous destination the Routing and RFID Agents are responsible. These agents are modelled similar to the approach already described in (Seitz et al., 2009). Additionally, the RFID Agent writes and reads all process steps needed to be accomplished on a RFID-tag placed on each Workpiece. This helps to track the production status of each Workpiece, even beyond different production locations or possible error states.

Our production system model, as shown in Fig. 2 is able to emulate a typical production setting as known from real production plants, down to its 24 volts industrial-grade sensors and actors. Every industry lane in the model is operated by five Gnublin embedded micro-controllers, supported by three RaspberryPi controllers for applications which need more resources. The Multi Agent System (MAS) Jadex is executed on a Siemens Industry PC to provide a platform where several software agents can be launched and work together.

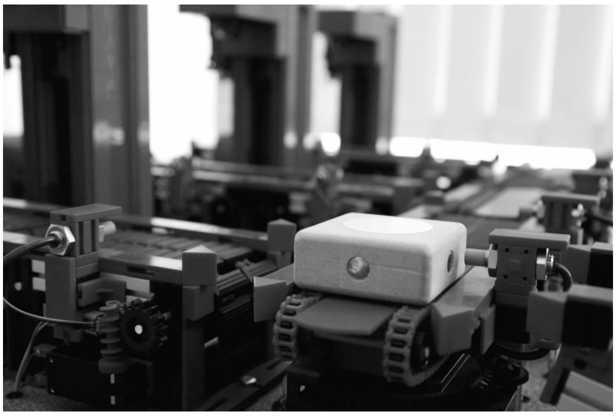

Fig. 2. Production system model

In order to face the challenges of these upcoming revolutionary forms of highly distributed industrial systems the German Ministry for Research and Technology founded the project Industry 4.0 (I4.0), which is also introduced the fourth industrial revolution. In cooperation with industries across Germany this research programme tries to push the integration of Information Technologies in the classic industry.

New forms of systems like the CPS and the Internet of Things drive these developments even further and are also used as fundamental sources for new innovative visions of future industry production scenarios. I4.0 aims for intelligent and fully automated systems which are able to make decisions on their own, depending on available resources and occurring events and provide vast new possibilities of organizing and executing production processes. This new form of industry makes implementation of individual, customer oriented processes more convenient (Industry 4.0 working group, 2013).

Systems implementing Industry 4.0 are often described as intelligent systems. Hereby, "intelligent" means, that the very products and decisions relating to the processes can be manipulated in real-time. These tasks are mostly accomplished self-organized utilizing technologies like software agents and Complex Event Processing (CEP).

Software Agent implementations can organize, plan, schedule and execute processes in industrial systems, without direct manipulation by or interaction with users. Implementing such software agents is a common way to build large autonomous systems, for example E-commerce or industrial systems.

CEP allows gathering and analysing important information in large, mass data producing systems. CEP engine rules can be defined, to filter the data stream and aggregate events accordingly. These events are used to trigger actions or receive information about the system's status.

An additional characteristic of Intelligent Systems is the usage of "Intelligent Products". These have the advantage that they can be identified and located at all times. They also give information about their current state and if there are possible, alternative ways to reach their strived, final state. These requirements can be implemented by using RFID[3]-technology for instance. Small RFID tags are attached to all products and resources in order to store information about these on a chip.

Machines and products gain intelligence by the use of sensors, the ability to be programmed and to be able to communicate with each other. That allows machine-to- machine communication within infrastructures. Here

[3] *Radio Frequency Identification*

machines are able to trigger other machine's actions, which render a new way of intelligent and autonomous production possible.

Our second test-bed CyPhREE (Cyber Physical [Objects for] Renewable Energies and Environment) is shown in Fig. 3.

CyPhREE provides simulation-based closed-loop control for smart buildings. Our CPS framework is used as foundation for the data acquisition, decision making and visualisation.

Smart buildings can use CPS with distributed sensors and actuators to interact with the physical world and get smart in this way. The sensors of a smart building are producing large amounts of data i. e. temperature or humidity readings, which have to be processed. This data integration can again be carried out with a combination of CEP and intelligent software agents. The CPS technology can be used to analyse events and coherences between them to suitably react on detected situations. The different software agents are used to fulfil the various jobs of the system and to encapsulate responsibilities. An example for a distinct job is the communication with the sensors and actuators of the smart building. The sensor of a photovoltaic power plant could indicate a high performance and the CPS could use this energy to activate the heating via an actuator.

The system architecture as described in "Concept and Design of a Cyber-Physical System for Smart Buildings" (Kögel, 2015) and "Cyber-Physical Systems for Smart Buildings" (Kögel, 2016) is shown in Fig. 4. Different sensors are sending their data via MQTT to the MQTT message broker (Fig. 4, step 1). The protocol is used for machine-to-machine (M2M) communication and represent a lightweight implementation of the publish/subscribe pattern. The MQTT software agent gets the sensor data from the MQTT server (Fig. 4, step 2) and interacts (Fig. 4, step 3) with the complex event processing software agent (CEP agent), which processes the incoming messages and analyses them. It can detect coherences of events and react on recognized situations.

Therefore it gives the operator software agent (Operator agent) a goal, which it will try to fulfil (Fig. 4, step 4). Each job is a separate goal, for example to save relevant data in the historical database (Fig. 4, step C1). On its way to the goal, it can interact (Fig. 4, step 5) with other agents like the building information model (BIM) software agent (BIM agent) to address the right actuators. The BIM agent knows all available sensors and actuators of the building and how to communicate with them (see (Kögel, 2015)). The operator agent will interact with the MQTT software agent to notify an actuator via MQTT (Fig. 4, step 6). It publishes the message on the MQTT server and the actuator, for example the HVAC system (heating, ventilation, air conditioning), will complete the goal (Fig. 4, step 7 and 8).

The BIM agent, introduced in (Kögel, 2016), also communicates with a RESTful web server (Fig. 4, step X and Y), which allows to receive metadata of sensors from a BIM (Fig, 4, step B). For example, a sensor could use the web service (Fig. 4, step Z and W) to get its MQTT topic on the broker to know where it has to publish the measured sensor data. This enables the ability to configure the distributed system in one place: the BIM. Further information on how to design BIM-based web services can be found in (Isikdag, 2015).

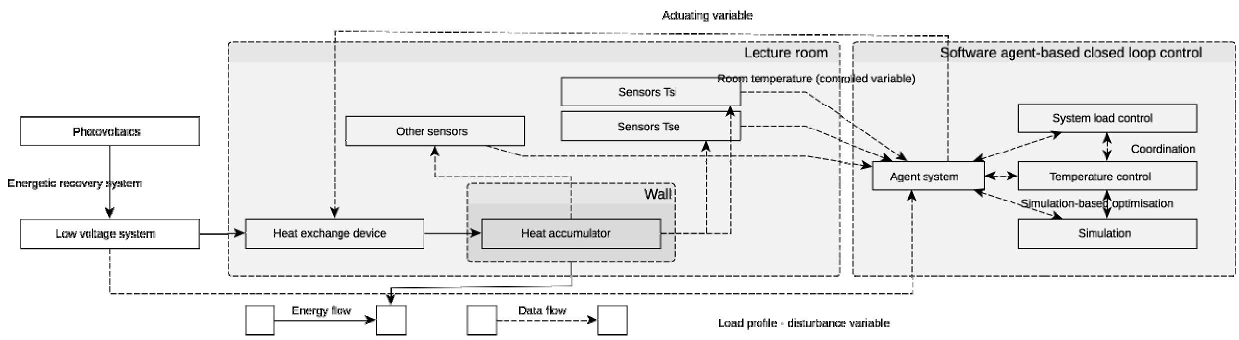

Fig. 3. CyPhREE system architecture

Furthermore, a simulation of the building is used to calculate the heat capacity of the building walls. For the simulation the surface temperature of a wall on both sides is measured and aggregated by the software agents. These mean values get published on the MQTT server and are used by the simulation. Afterwards, the calculated heat capacity of the wall is published back to the software agents. There it can be used for the heating of the building. In this way, excess energy could be dissipated, to activate the walls of the building and use them as a thermal storage system. To complete this, the floor plan should be analysed, to heat occupied rooms with available heat capacity predictive. (Bauer et al., 2016).

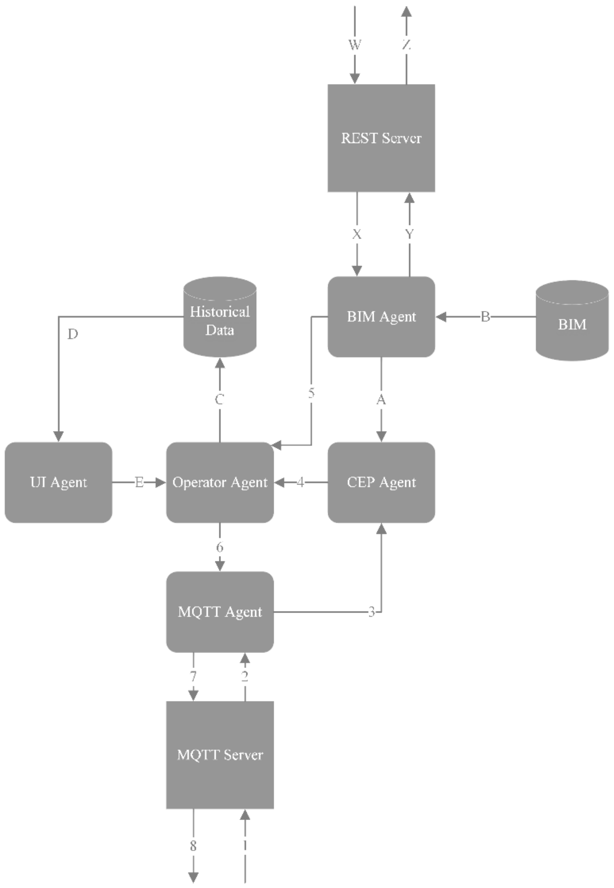

Fig. 4. CyPhREE software agents (Kögel, 2016)

DATA SCIENCE AND THE INTERNET OF THINGS

The term data science has been coined some 40 years ago. It was synonym for today understands of computer science. Today data science describes means of managing and scientific processing vast amounts of data. For some time now, the term is also known as big data. In big data, there are three dimensions of data: (1) data volume, (2) data velocity, and (3) data variety.

Data volume (1) describes the property that is mostly connected with big data: Vast amounts of data, not only megabytes but petabytes of data. Data velocity (2) describes the rate where data needs to be processed. Big data is not only concerned with batch processing but also with real-time data acquisition and processing. Finally, data variety (3) describes the fact, that data is no longer only provided in a strongly structured way (e. g. data tables in RDBMs) but more in more in unstructured data streams like log files, twitter feeds, etc. In our application examples we are currently facing small data volumes (some kilo- or megabytes of data) with high velocity (event-based data streams) and medium data variety (semi-structured events).

From a data-flow point of view, our approach is structured according to the JDL model for data fusion (see Fig. 5 for details).

DATA FUSION

The integrated model for data fusion describes to data flows, one for data fusion and another one for data mining. The data fusion flow describes on level 0 the acquisition of sensor data into object information. On level 1, the object information is further refined into information about the situation the system is currently in. On level 2 this situation information is processed and visualised. Finally, on level 3, impacts of the current situation are derived, from which actions can be taken, either manually or automatically.

DATA MINING

Additionally the JDL model for data fusion describes a data flow for data mining. Acquired data is persistently stored in a data warehouse. After data cleansing and necessary transformations the data can be fed into data mining frameworks to gather new knowledge from it. The new knowledge is stored in some sort of model (e. g. explicit knowledge model (ontology) or statistic weights (e. g. neural networks)).

From the knowledge model, gathered from data, one can generate new patterns for object or situation detection in the data fusion flow (level 4, resource refinement).

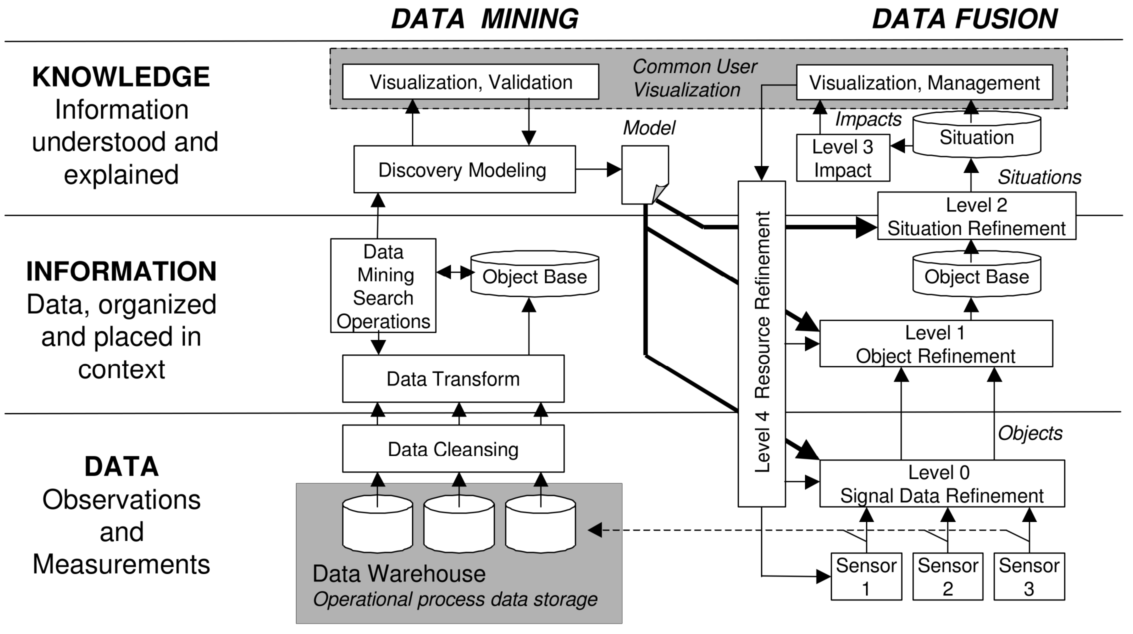

Fig. 5. Integrated model for data fusion and data mining (Waltz, 1998)

EXAMPLE DATA FUSION APPLICATIONS

In our factory automation example, sensor data is generated from various industrial sensors as well as from RFID sensors. An example pattern for object detection can be the presence detection of a Workpiece via a digital input sensor in combination with the read-out of the RFID tag on the Workpiece. Both sensor events together enable the detection of a Workpiece object. Two adjacent digital sensors on a conveyor may signal the exit of a Workpiece from one conveyor position segment, as well as the entry into another conveyor position segment in the factory. Both sensor events together (following each other in a short time frame) may indicate the movement of a Workpiece on the conveyor (situation detection). Knowing all Workpiece movements in the factory, one is able to predict expectable Workpiece stalls on particular conveyors (impacts). In our example applications, the object/situation/impact detection is carried out using Complex Event

Processing (CEP), a method for processing large amounts of sensor events via declarative patterns in an Event Processing Language (EPL) (Seel et al., 2010).

In case of the building automation system, object/situation/impact detection is also carried out using CEP. For example, from temperature and humidity sensor readings, an intelligent CEP agent is able to establish a controlled “feel-good climate” for a particular lecture theatre at the university. Currently, simple hystereses as well as PID closed-loop control algorithms are implemented in EPL statements.

DATA MINING APPLICATIONS

For both application scenarios (factory and building automation) raw event and derived event data is stored in a time series database system (i. e. data warehouse). The data can be fed to various data mining or deep learning frameworks to classify particular situations from provided data. In the factory automation use-case, the data and neural networks will be used to optimise the sequence of the production of Workpieces in order to minimise setup times.

Furthermore, the gathered sensor and RFID data will be used for predictive maintenance. From variations in event flows, we will learn certain factory conditions (e. g. OK, maintenance needed, stalled, etc.) which in turn can be used to predict failures in production processes.

In the building automation use-case, we are currently investigating, whether the time series data (temperature, humidity, etc.) can be used to learn situations such as changes in ventilation (window open/closed) or patterns of room usage (lecture, workshops, etc.) from the gathered data.

In both scenarios, the learned patterns will be fed back to the data fusion data flow to improve object/situation/impact detection.

Table gives an overview on used software and frameworks for both data fusion and data mining.

Table. Software stacks for data mining and data fusion

Data mining	**Data fusion**
Jupiter notebook	Cross-platform/web user interface
Deep Learning (e. g. PyBrain, Blocks, Lasagne, Neuroph, etc.)	Artificial Neural Networks, CEP, Haddon, Spark, etc.
TensorFlow, Theano, Spark, Hadoop, etc.	Multi agent system
Ducker	Java/Python runtime
Linux	Embedded Linux

CONCLUSIONS

We have shown, that applications in the are of the Internet of Things, particular smart factory and building automation via cyber-physical systems is a sound foundation for data science applications. We have described our concepts and frameworks used in productive set-ups for both use-cases. Currently we are investigating the application of concepts and methods from deep learning in our scenarios.

REFERENCES

1. Bauer M., Schöler T., Kögel L., and Schweinfest M., (2016), Agentenbasierte Steuerung von thermischer Simulation in Echtzeit, *Bausim 2016, Dresden, Sep. 6 (to be published).*

2. Industrie 4.0 working group, Recommendations for Implementing the Strategic Initiative Industries 4,04.0. 2013.

3. Isikdag U., (2015), Advanced SOA Patterns for Building Information Models, in SpringerBriefs in Computer Science, Enhanced Building Information Models: *Using IoT Services and Integration Patterns*, U. Isikdag, Ed. 1st ed, Cham, s. l.: *Springer International Publishing,* pp. 71 – 98.

4. Kögel L., (2015), Concept and Design of a Cyber-Physical System for Smart Buildings, *in Applied Research Conference 2015*, Nürnberg, book-on-demand Verlag, pp. 72 – 75.

5. Kögel L., (2016), Cyber-Physical Systems for Smart Buildings, *in Applied Research Conference 2016, Augsburg,* (to be published).

6. Schöler T., (2014), The SEPIA cyber-physical production control system, *Electrotechnic and Computer Systems, ONPU* Odessa, Ukraine, No. 13 (89),

7. Schöler T., Ego C., Leimer J., and Lieback R., (2013),Von Softwareagenten zu Cyber-Physical Systems: Technologien und Anwendungen, in Agentensysteme in der Automatisierungstechnik, P. Göhner, Ed. Springer Berlin Heidelberg, pp.129 – 149.

8. Seel C., Schimmelpfennig J., Mayer D., and Walter P., (2010), Conceptual Modelling of Complex events of the Internet of Things, *in eChallenges, 2010,* , pp. 1 – 8.

9. Seitz C. and Schöler T., (2009), An Agent-based Sensor Middleware for generating and interpreting Digital Product Memories.

10. Vogel-Heuser B., Schütz D., Schöler T., Pröll S., Jeschke S., Ewert D.,. Niggemann O, Windmann S., Berger U., and. Lehmann C, (2015), Agentenbasierte Cyber-physische Produktions-systeme, atp Edition, Bd. 57, No. 9/2015, pp. 36 – 45.

11. Waltz E.L., (1998), Information Under-standing: Integrating data Fusion and data Mining Processes, *in Proceedings of the 1998 IEEE International Symposium on Circuits and Systems, ISCAS '98*, Bd. 6, Vol. 6, pp. 553 – 556.

Received 25.05.2016

Thorsten
Schöler, Prof. Dr.-Ing.
Augsburg University of Applied Sciences, Augsburg, Germany.
E-mail: thorsten.schoeler@hs-augsburg.de

Lucas
Kögel, BSc, MSc (cand.),
Augsburg University of Applied Sciences, Augsburg, Germany.
E-mail: lucas.koegel@hs-augsburg.de

Sebastian
Pröll, BSc, MSc (cand.),
Augsburg University of Applied Sciences, Augsburg, Germany.
E-mail: sebastian.proell@hs-augsburg.de