UDC 004 932.72'1

T. Schoeler, Dr.-Ing.

THE SEPIA CYBER-PHYSICAL PRODUCTION CONTROL SYSTEM

Abstract. *The article introduces a test-bed for research cyber–physical production systems. The hardware setup and the software environment are described. A new approach for distributed software and semantic data modelling is introduced.*
Keywords: *cyber-physical systems, embedded production control, semantic data model, 3D visualisation*

Т. Шёлер, д-р техн. наук

КИБЕР-ФИЗИЧЕСКАЯ СИСТЕМА УПРАВЛЕНИЯ ПРОИЗВОДСТВОМ SEPIA

Аннотация. *В статье представлена модель для исследования кибер-физических систем в области промышленного производства. Выполнено описание процесса установки оборудования и программной среды. Представлен современный подход для моделирования распределённого программного обеспечения и семантических данных.*
Ключевые слова: *кибер-физические системы, встроенное управление производством, семантическая модель данных, 3D визуализация*

Т. Шьолєр, д-р техн. наук

КІБЕР-ФІЗИЧНА СИСТЕМА УПРАВЛІННЯ ВИРОБНИЦТВОМ SEPIA

Анотація. *У статті представлена модель для дослідження кібер–фізичних систем у галузі промислового виробництва. Виконано опис процесу встановлення обладнання та програмного середовища. Представлено сучасний підхід для моделювання розподіленого програмного забезпечення і семантичних даних.*
Ключові слова: *кібер–фізичні системи, вбудоване управління виробництвом, семантична модель даних, 3D візуалізація*

Introduction. The German research and development initiative for Industry 4,0 proposes a highly networked and more flexible ecosystem for modern production control systems [1].

To conduct research in the area of flexible production control systems, a flexible test–bed installation is a valuable support. An example for a flexible plug-and-produce scenario is given in the well-known Production 2000+ agent-based manufacturing system [2].

Our own research has already shown, that traditional production control systems, based on programmable logic control (PLC) and Object-Linking and Embedding (OLE) for process control, are not able to deal with the emerging requirements imposed on modern production control systems by rising flexibility and autonomy, like self-configuration and self-optimisation. These self-x properties are not new and have already been described in the autonomic computing research field e. g. by IBM in [3].

Currently these described research efforts are subsumed in the term Cyber-Physical Production System (CPPS) (Source: http://www.plattform-i40.de/glossar/cyber-physical-production-systems-cpps).

The remaining sections of these contributions are structured as follows. First, an overview over our SEPIA CPPS test-bed is given. Subsequently some aspects of our software architecture are described in more detail, namely the extension to the Contract Net Protocol and our semantic data model for information storage, the production ontology. The test-bed and its software architecture is then used in two exemplary applications for event processing and factory visualisation in 3D. Finally, this contribution describes current research work and future perspectives.

System SEPIA – a semantic cyber–physical system approach.

System SEPIA is our research test-bed for a state-of-the-art production control system. It is built from competitive off-the-shelf hardware and software components. Fig. 1 shows the hardware simulation model of a realistic production scenario built from Fischer technikcal components.

The model reproduces a complex manufacturing scenario with three manufacturing cells, conveyors for work piece distribution as well as storages.

The manufacturing model is automated either with traditional PLC connected via industrial Ethernet (Profinet) and to provide the needed flexibility for research, with off-the-shelf industrial-grade embedded controllers as shown in Fig. 2.

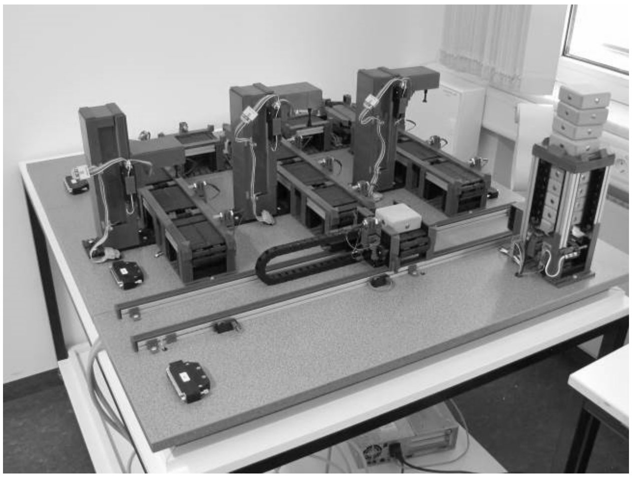

Fig. 1. Hardware simulation model

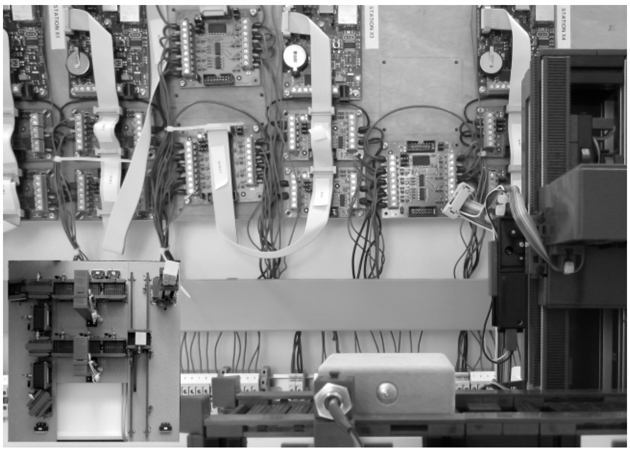

Fig. 2. Industrial–grade embedded controllers for hardware simulation model

On field level five Gnublin LAN embedded Linux controllers are used to connect to sensors and actors of the plant model. The used Gnublin controllers contain an ARM9 CPU at 180 MHz with 32 MB SDRAM and dedicated hardware to work with 24 V digital input and outputs (sensors and actors in the model). The Gnublin controllers are all interconnected via industrial Ethernet.

Gnublins provide a standard Linux environment for software development. They can easily be programmed with programming languages like C/C++ and Python. Furthermore convenient application programming interfaces (APIs) are available to implement sensor/actor access for industrial control systems.

To realise the various functionalities traditionally found on ANSI/ISA-95 (more information available from http://www.isa-95.com) levels 2 and 3 (higher control and manufacturing execution levels) SEPIA uses a set of Raspberry Pi controllers. These controllers are also based on embedded Linux but more powerful than the field-level Gnublin controllers. The Raspberry Pi and Gnublin controllers are interconnected via industrial Ethernet as shown in Fig. 3.

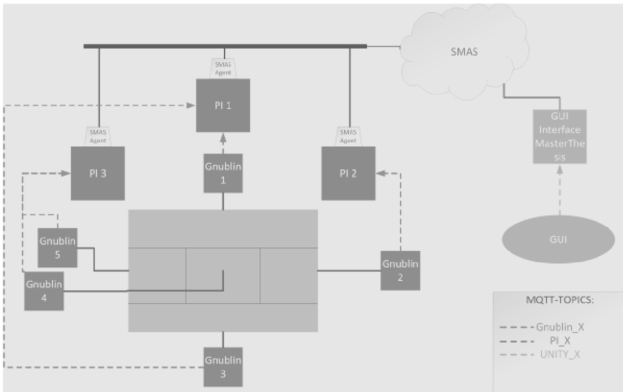

Fig. 3. Controller architecture

As already described, the various embedded controllers in SEPIA system are connected via industrial Ethernet. On top, SEPIA uses MQTT for message passing.

Message queuing telemetry transport (MQTT) is a M2M (Machine to Machine) communication protocol built for the Internet of Things: "MQTT is a lightweight publish/subscribe protocol flowing over TCP/IP for remote sensors and control devices through low bandwidth, unreliable or intermittent communications." as described in [4].

In SEPIA the lower field level controllers communicate via MQTT, whereas on higher levels of manufacturing control, more sophisticated communication means are necessary. Therefore SEPIA uses Scala Multi Agent System (SMAS). More information available at URL: https://github.com/scala-multi-agent-system for higher-order communication and control. SMAS is written in the object-functional programming language Scala. It uses the actor concept for message passing between software agents and provides services similar to well-known software agent platforms such as

JADE (Java Agent Development Framework.) More information available at URL: http://jade.tilab.com and Cougaar URL: http://cougaar.org/wp/[5].

SMAS enables agents to communicate with each other, for example as proposed in the Contract Net Protocol for task sharing and resource allocation in distributed environments [6]. We propose an enhancement of CNP, the so-called RT-Contract-Net optimised specifically for non-isolated, non-homogeneous real-time environment such as SEPIA.

RT-Contract-Net provides a channel system for service discovery as well as intelligent message routing based on the messages' real-time requirements and real-time capabilities of the participants. Furthermore, it provides various means for optimisation as shown in Fig. 4.

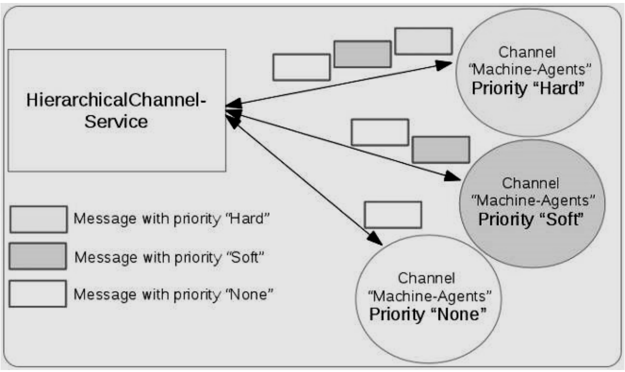

Fig. 4. RT-Contract-Net's hierarchical channel service and message routing

The hierarchical channel service in RT-Contract-Net distributes Contract-Net Protocol messages (CNP) messages only to the participants known to provide suitable real-time capabilities. For example, messages with hard real-time requirements are only distributed to participants providing hard real-time capabilities, whereas messages with no real-time requirements are distributed to all participants in the network. More background on RT-Contract-Net can be found in [7].

Semantic data model.

To provide the various software agents with knowledge about the production plant model itself and for manufacturing execution and production control, an ontological description logic model in Web Ontology Language Description Logic (OWL DL) is proposed for SEPIA. Similar approaches can be found e. g. in [8]. An overview of the ontology and its main entities is given in Fig. 5.

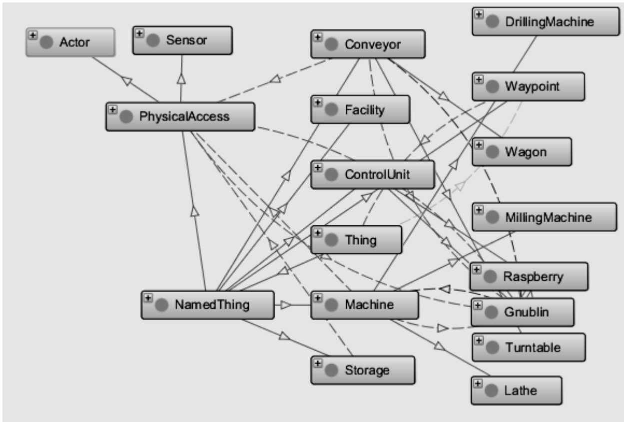

Fig. 5. SEPIA ontology entities

The SEPIA ontology models the hardware parts in the plant like e. g. conveyors, machines, storages as dedicated entities in a subsumption hierarchy. Also the embedded controllers are modelled in the ontology as control units. Given this logical model of the production plant hardware setup, software agents are able to reason about the hardware setup.

For example a configuration software agent is able to derive particular configuration settings for the various Gnublin embedded controllers from the ontology. Information such as which controller is attached to a particular machine or conveyor as well as how to access the various sensors and actors in the production control system can easily be accessed via the SEPIA ontology.

Furthermore due to the graph structure of the production plant, a simple relation such as *hasNeighbour* can provide a routing software agent with information necessary to optimise the transport routes through the production plant. An example for an ontology class model is given in Fig. 6.

```
hasAddress some string
isConveyorOf only ProductionLane
hasNeighbour some
    (Conveyor
     or Machine
     or Storage)
hasPhysicalAccess some PhysicalAccess
isControlledBy some Gnublin
```

Fig. 6. Conveyor class, modelled in OWL DL ontology

The conveyor class provides properties like *hasNeighbour* and *isControlledBy*. The *hasNeighbour* relation provides adjacent production plant entities, the *isControlledBy* property provides information about the embedded controller in charge of controlling the given production plant entity.

By using the query language SPARQL (SPARQL Protocol And RDF Query Language), software agents can obtain information from the ontology necessary for e. g. configuration of embedded controllers. Fig. 7 shows an example query.

```
SELECT ?x ?subs WHERE {
    Type(?x, cpsi:ControlUnit),
    PropertyValue(?x,
        cpsi:hasIdentification, id),
    PropertyValue(?x,
        cpsi:isSubscribing, ?subs)
}
```

Fig. 7. Example of a SPARQL query

This example queries all MQTT topics, a particular embedded controller is subscribed to, as stored in the production control ontology.

Currently the Scala multi agent system SMAS is enhanced with plug-ins to access OWL DL ontologies. The plug-ins provide direct manipulation of the OWL DL ontology as well as querying via SPARQL as needed in the SEPIA production control system. SMAS agents are deployed on Raspberry Pi nodes for higher-order production control and manufacturing execution. Due to resource constraints on the limited Gnublin nodes, a OWL DL ontology proxy agent is provided and deployed on a Raspberry Pi node. The proxy agent is accessed via MQTT messages, provides access to the ontology and posts its results back as MQTT message. This enables field level controllers to store current production data in the ontology and also retrieve e. g. configuration data from the production ontology.

Event correlation.

Data stream management and complex event processing technologies are well-suited for real-time and online analysis of complex situations. Especially in logistics scenarios such as food chain logistics, online decisions are necessary to optimise food handling throughout the delivery to the consumer [9].

A similar data flow concept can be found in SCADA (Supervisory Control and Data Acquisition) and quality management applications in the manufacturing control. Sensor information is gathered on field level and is correlated and evaluated on a higher level in the architecture. Result of correlation and evaluation steps is e. g. an alarm, when certain alarm triggers are met.

In SEPIA, the correlation and evaluation step is carried out with complex event processing technologies. Therefore a Scala Multi Agent System (SMAS) software agent has been deployed, which is able to correlate higher order events, such as alarms or visualisation events from field level sensor and actor events. As a very simple example, to explain the general idea, a workpiece leaving a particular conveyor (registered by a sensor event *E1* on conveyor 1) and entering another plant sector (registered via another sensor event *E2* on conveyor 2) can be shown on the visualisation layer by a moving workpiece.

For quality management, manufacturing events from the manufacturing cells, such as a workpiece being manufactured with a particular tolerance, can also be correlated and evaluated by the event processing agent. In case, that the manufactured workpiece is out of the given tolerance, the quality management event processing agent is able to trigger, e. g. a rework process for the particular workpiece.

3D plant visualisation.

The 3D plant visualisation data flow is given in Fig. 8.

Fig. 8. Data flow visualisation

Field level sensors and actors provide their data via MQTT to an event correlation agent, which in turn correlates sensible visualisation information from the low-level MQTT events and provides these higher-order events via MQTT or SMAS messages to the 3D visualisation on desktop PCs or on mobile devices.

According to [10] a 3D visualisation has additional value if the user is able to interact with the visualisation. Therefore we propose a software architecture for event-driven 3D visualisation using the Unity 3D framework (More information available from http://unity3d.com) as shown in Fig. 9.

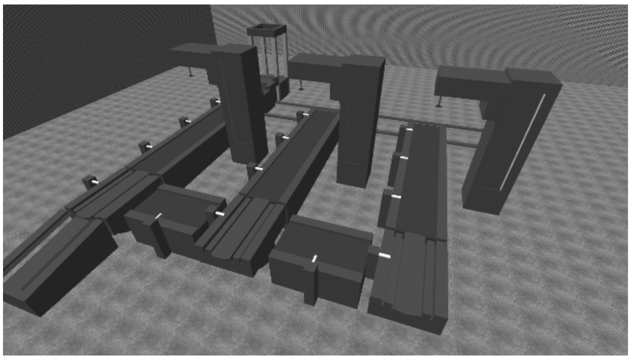

Fig. 9. Animated 3D visualisation of plant model

Apart from Unity's support for realistic 3D visualisation, together with shutter glasses, a realistic 3D impression can be generated with regular desktop PC hardware. Furthermore, Unity allows cross-platform developments. The same code base can be used for supervisory control and data acquisition (SCADA) on desktop PCs as well as on mobile devices, such as Android tablets.

Current developments.

Currently, the plant model only provides workpiece IDs via read-only magnetic markers on the workpieces. As proposed in various other projects, e. g. SemProm [11], Radio Frequency IDentification (RFID) provides contact-less identification of workpieces and furthermore, allows not only to read a given ID but also to write data on RFID tags. This data can be used to store e. g. work orders, quality management information directly on the workpiece itself. RFID technology can easily be integrated in the SEPIA test-bed hardware and software architecture. Therefore dedicated MQTT topics will be provided by Gnublin controllers to read/write RFID tags.

One important property of cyber-physical production systems is the high degree of distribution of the participants in the system. Therefore communication and coordination between distributed manufacturing plants is an important research topic. To demonstrate such a distributed manufacturing scenario, a highly distributed demonstrator is currently developed by a number of German research institutes: The Industrie 4.0-Demonstrator: MyJoghurt [12]. System SEPIA testbed will be integrated into the MyJoghurt demonstrator as another participant.

A dedicated Scala Multi Agent System (SMAS) plug-in, encapsulating the C interface code will be used to provide the external interface to the other participants in the demonstration scenario.

Conclusion

We have introduced the SEPIA cyber-physical production system test bed as well as its hardware and software architecture.

Current research has shown, that a lot of research in the field of software agent theory can be used to build state-of-the-art production control systems. Some enhancements, e. g. in the field of task sharing (Contract Net Protocol), have been proposed in order to deal with the emerging requirements from highly distributed and connected real-time systems such as systems driving the Internet of Things.

We have shown that a semantic data model kept in an Web Ontology Language Description Logic (OWL DL) ontology has numerous advantages in knowledge modelling, especially when it comes to enhance to data model to cope with additional requirements from future applications.

Furthermore, technologies from data stream management systems can be used in manufacturing systems for visualisation and quality management. Those technologies can easily be integrated in the open SEPIA software agent system architecture.

The integration in the German Industrie 4,0 demonstrator MyJoghurt provides a reality-check for the proposed architecture and its components, bridging the gap between academic research and productive use in realistic production scenarios.

Received 28.02.2014

References

1. Industrie 4.0 working group. Recommendations for Implementing the Strategic Initiative Industrie 4.0, 2013. URL: http://www.acatech.de/fileadmin/user_upload/Baumstruktur_nach_Website/Acatech/root/de

/Material_fuer_Sonderseiten/Industrie_4.0/Final_report__Industrie_4.0_accessible.pdf.

2. Bussmann S. Self-Organizing Manufacturing Control: An Industrial Application of Agent Technology, (2000), *In Proc. of the 4th International Conference on Multi-Agent Systems (ICMAS'2000).* Boston, URL: http://www.stefan-bussmann.de/downloads/icmas00.pdf.

3. Murch R. Autonomic Computing, (2004), *Upper Saddle River N.J. Prentice Hall Professional Technical Reference.*

4. MQ Telemetry Transport (MQTT) V3.1. *Protocol Specification. CT316*, August 19, 2010. URL: http://www.ibm.com/developerworks/webservices/library/ws-mqtt/.

5. Lieback R. Objekt-funktionale Plattform für Cyber-Physical Systems. Bachelorarbeit, (2012), *Augsburg: Hochschule für Angewandte Wissenschaften.*

6. Smith R. G. The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver. *IEEE Transactions on Computers,* 29 p., No. 12 (December 1980): pp. 1104–1113. doi:10.1109/TC.1980.1675516.

7. Wiedenmann F. Eine Contract-Net- und Multi-Agenten-System-Erweiterung für Echtzeitumgebungen. Masterarbeit, (2014), *Augsburg: Hochschule für Angewandte Wissenschaften.*

8. Georgoudakis M., Alexakos C., Kalogeras A., Gialelis J., and Koubias S.. Decentralized Production Control through ANSI, (2006), *ISA-95 Based Ontology and Agents. In 2006 IEEE International Workshop on Factory Communication Systems*, pp.374 – 379, doi: 10.1109/WFCS.2006.1704182.

9. Alexander F., Hipp T., Scholze T., Wagner F., Schöler T., and Krupp M. Das dynamische Mindesthaltbarkeitsdatum – Auf dem Weg zu einer Echtzeitereignisverarbeitung in der Lebensmittellogistik. *Magdeburg*, 2013. URL: http://www.btw-2013.de/proceedings/Das%20dynamische%20Mindesthaltbarkeitsdatum%20%20Auf%20dem%20Weg%20zu%20einer%20Echtzeitereignisverarbeitung%20in%20der%20Lebensmittellogistik.pdf.

10. Back M., Childs T., Dunnigan A., Foote J., Gattepally S., Bee Liew, Jun Shingu, and Vaughan J. The Virtual Factory: Exploring 3D Worlds as Industrial Collaboration and Control Environments, (2010), *In 2010 IEEE Virtual Reality Conference (VR),* pp. 257 – 258, doi:10.1109/VR.2010.5444777.

11. SemProM – Foundations of Semantic Product Memories for the Internet of Things. Accessed February 20, 2014. URL: http://www.springer.com/computer/ai/book/978-3-642-37376-3.

12. Lehrstuhl Für Automatisierung und Informationssysteme: I4.0-Demonstrator: MyJoghurt. Accessed February 20, 2014. URL: http://www.ais.mw.tum.de/i40-demonstrator-myjoghurt/.

Schoeler Thorsten, Prof. Dr.-Ing., Works at Augsburg University of Applied Sciences, Germany. His research interests are Distributed Systems, Especially In industrial Applications as well as Mobile Computing.
E-mail: thorsten.schoeler @ hs-augsburg.de.
Telefon: +49 (0)821 5586-3445
Friedberger Straße 2,a
86161 Augsburg