# Bridging the Gap between Digital Object Memory Information and Its Binary Representation

Marc Seissler
University of Kaiserslautern
Kaiserslautern, Germany
marc.seissler@mv.uni-kl.de

Ines Heck
SmartFactoryKL
Kaiserslautern, Germany
heck@smartfactory.de

Peter Stephan, Jochen Schlick
German Research Center for Artificial Intelligence
Kaiserslautern, Germany
{peter.stephan, jochen.schlick}@dfki.de

## ABSTRACT

Today, several applications in the domain of industrial production and logistics use information stored on smart labels to control processes or monitor the flow of goods. This often complex information is represented by bits and bytes e.g. on a RFID transponder and is usually modeled accordingly, by arranging the bits and bytes in an order suitable for the specific application at hand. Accessing the information means coding this order into read/write algorithms in PLCs or other automation systems, making it difficult and expensive to add, rearrange or remove pieces of information. To increase the benefit of IoT technologies and to allow a seamless migration of the underlying tag technologies, future systems thus have to abstract from the information and its binary representation.

In this paper we introduce a mapping technology that enables the separation of information and its' Digital Object Memory (DOM)-specific memory representation. The key of this concept is an XML schema used for the separated description of information and its DOM-specific memory address. A server has been developed that stores those tag-specific XML documents and supports querying for the requested information addresses. To demonstrate the feasibility of the approach this mapping technology has been used in combination with a programmable logic controller (PLC) to enable the flexible migration of various smart label technologies and DOM representations avoiding the necessity to change the PLC application code.

## Categories and Subject Descriptors

J.7 [**Computers in other Systems**]: Industrial Control; K.4.3 [**Organizational Impacts**]: Automation; D.2.12 [**Interoperability**]: Data Mapping

## General Terms

Design, Experimentation, Verification

## Keywords

Digital Object Memory, Internet of Things, Distributed Systems, Radio Frequency Identification, XML


*UbiComp 2010*, Sep 26 – Sep 29, 2010, Copenhagen, Denmark.


## 1. INTRODUCTION AND MOTIVATION

Looking at today's economy, it becomes obvious that industry is facing a range of unprecedented challenges. Globalization, shortening product lifecycles and an increasing customer desire for individualized goods are asking for highly flexible processes in the production and logistics industry which are able to adapt quickly to changing market demands [20]. First solutions to meet these challenges are provided by technologies for the "Internet of Things" [5] (e.g. barcode and radio frequency identification (RFID)), bridging the gap between the physical flow of goods and the digital flow of information [1, 4].

A more comprehensive approach to associate digital lifecycle information with physical objects is represented by the concept of so-called digital object memories (DOM) [7, 14]. As it has been shown in [10], DOM goes beyond current real life Auto-ID applications and allows for an efficient and flexible exchange of product lifecycle information between varying stakeholders of a value chain [15]. By that, the modeling of the on-product DOM format becomes an important factor for the operability of a developed solution [10]. DOM information stored on a product can be directly transferred as binary code into the programmable logic controller (PLC) of factory modules for parameterizing production processes and by that, support a decentralized control strategy [15].

In order to make the industrial application of DOM a success, two main problems need to be addressed. First of all, concepts need to be derived to allow industry adopters to easily migrate from state-of-the-art Auto-ID technology to the initial level of commercially viable DOM. That includes shifting from concepts mainly based upon unique IDs referencing information in backend systems with high computing power to DOM allowing for true on-product information storage (e.g. so called Storage DOMs based upon passive RFID tags). This is not trivial, as in comparison to more autonomous but expensive DOMs as presented in [13], storage DOMs allow for reading and writing data directly from/onto an object, where else it does not possess further intelligence as opposed to Smart and Autonomous DOMs which can be directly accessed by a semantic or block interface. In comparison to that Storage DOMs have to be decoded in order to access the meaning of the stored data (see Figure 1).

In a second step, concepts are required to enable industry users to quickly adapt to changing binary formats and technological changeovers of memory hardware like smart labels or RFID transponder types [15]. In current systems the knowledge about addresses under which process-relevant DOM information is stored is an implicit part of the PLC application code. This means in case of changes in DOM format or technology leading to alterations in memory information addresses, a recoding of all PLCs involved in a production process is the consequence. Therefore solving this expensive, time-consuming and resource-intensive issue is crucial for a successful application of DOM in industrial application domains.

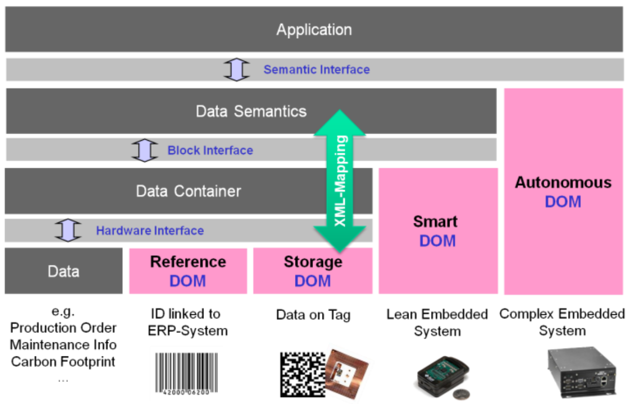


**Figure 1. Semantic Gap between Storage DOMs and their Information**

The solution presented in this paper shows a technical approach to the problem of hard coded bit addresses in DOM and the related issue of inflexibility in the area of industrial automation. As storage DOM comes closest to the technology currently in use by the production industry, our approach covers this kind of memory technology. In the center of the our concept stands an easily maintainable XML document that specifies bit addresses of DOM information which is stored in an on-product memory and replaces static lookup tables in the PLC code. To evaluate the feasibility of the chosen approach, the developed components are implemented in a real life production system consisting of a PLC, RFID read/write devices and industrial control software. Based on the experiences gathered during comprehensive tests, benefits and limitations of the current solution are discussed and future research directions are derived.

## 2. STATE OF THE ART

Technologies like barcode or RFID systems are an integral component of today's factory and automation systems. Major examples for the use of Auto-ID systems can be found in various domains like production, logistics and retail, ranging from RFID-based production systems [18], via automated airport baggage handling systems [8] to the tracing of products in the retail sector [6]. Hardware technologies are complimented by ID consortiums like the electronic product code (EPC) or the GS1 coding system, referencing backend information by using an object's ID [3, 12]. Most of current solutions use Auto-ID technologies and object-related information only in the context of closed-loop applications in single domains or even a single company [9]. Nevertheless the idea of using decentralized data within the context of production process control is already around for some time [9, 18].

In order to provide current automation systems with a higher degree of flexibility, more and more mature IT technologies move into the domain of factory automation. Examples are communication technologies like WLAN and Bluetooth or mobile devices like smartphones and wearable computers [19].The same accounts for the use of DOM in industrial application domains.

When looking at past and ongoing research activities in information technology, examples for DOM are manifold [11, 13, 14, 17].

What sets the use of DOM in automation systems apart from other application domains is that information must be dealt with on a binary level. The inability to do computing on a higher level within a commonly used PLC and the still very limited storage space of current transponder technologies makes it difficult to use textual data formats – like XML – on the DOMs. Because of this constraint an efficient binary coding of the DOM data format becomes highly relevant to provide a solution that is expressive and flexible enough to deal with comprehensive life cycle information [15].

In order to use product- and production-related information for process parameterization in open-loop factory processes, an on-product memory solution was developed in the project *Semantic Product Memory* (SemProM) and evaluated in real-life demonstration systems [10]. Nevertheless the problem of interpreting binary data in order to parameterize or to control a process remains. Since the interpretation of the information is still implemented on the PLC level the interpretation knowledge is encapsulated within the code and inflexible with respect to a change of the DOM structure.

In order to realize a more flexible handling of binary DOM information in factory automation systems, web technologies seem to provide a remedy. The eXtensible Markup Language (XML) [16] is a declarative description language that is broadly used for the platform-independent specification of data formats in today's information systems. To use these technologies within automation systems, an efficient binary representation for the XML documents is demanded.

The Efficient XML Interchange (EXI) [2] format is a recommendation for the efficient exchange of XML documents. EXI can be used to transform XML documents in a compact binary representation which is suitable for a fast document processing. Since EXI encoder use entropy encoding for transforming XML information into a binary XML representation, the structure of the output stream may vary, depending on the information of the input document. But since the addresses of the information stored on a DOM are commonly fixed, the encoder has to map the input information onto their static addresses. Since EXI doesn't support this information mapping, it is not suitable to be used for the binary representation of XML documents on DOMs.

To enable a flexible, hardware independent use of DOM technologies in future automation systems the next section presents a mapping approach, which decouples the DOM information from its binary representation using XML technology.

## 3. APPROACH

In today's processes there is a "digital gap" between the DOM structure specification and the implementation of the DOM access routines on the hardware level. Informal specifications – like Excel sheets and Word documents – are commonly used to specify the memory addresses of the data stored on the DOM. These documents are then used by the programmers to implement the PLC routines which assign the DOM data to the variable used within the program. Usually a static lookup table is used to define the variable memory addresses on the DOM. Since the lookup table is fixed within the PLC code, the program would have to be reprogrammed and recompiled if a memory address is changed.

This static lookup table is replaced with a new lookup mechanism that allows the outsourcing of the memory addresses into an easily maintainable XML document.

As depicted in Figure 2, the system architecture consists of three main components: The first component is a programmable logic controller (PLC) which controls the automation process. As the first step within the process, the PLC uses the RFID read/write devices to read the DOM data into the internal memory buffer. Instead of using an internal lookup table to decode the variable values, a new module has been integrated in the PLC that queries the variable addresses on demand. If the memory address of a variable is unknown to the PLC, an external "mapping server" is called by sending the variable's namespace address. This namespace address is coded as an XML XPath String, which comprises the variable's position within the hierarchical DOM structure.

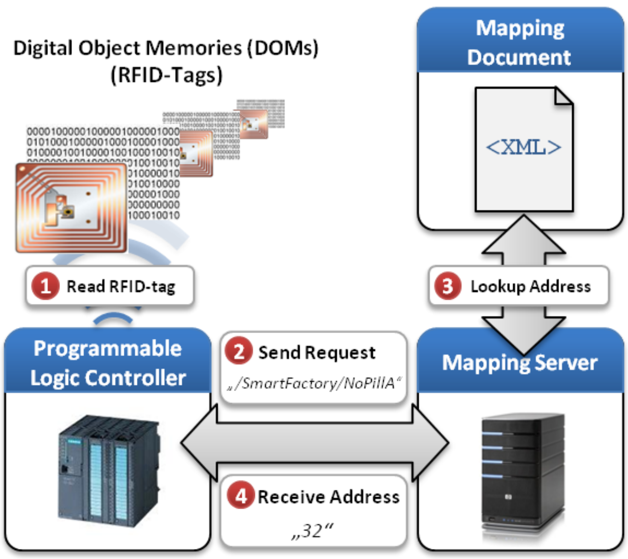


**Figure 2. The Architecture of the Mapping-Concept**

The DOMs hierarchical structure is specified using so called *blocks*. Blocks are logical structures that can be used to group information, e.g. by vendor, factory, production cell or production module.

For example the variable *NoPillA*, which is used within the SmartFactory block to indicate how many dietary supplement pills of Type A have been filled in a blister, is represented by the logical namespace "/SmartFactory/NoPillA" (see Figure 2).

The second module of our concept is represented by the mapping server which is used to process the variable requests that are sent by the PLC. The mapping server looks up the requested variable memory addresses in an XML document, which stores the variable memory addresses for a specific DOM. This XML document is the third component representing the core of the mapping approach. For each specific DOM structure, an XML mapping document is specified and stored on the server.

According to the DOM structure an XML *block*-element is used to hierarchically organize the variables in the XML document. The leaves of the XML document are represented by the atomic *blockpart* elements (see Listing 1). Each blockpart element describes one variable and its memory address on the DOM. The memory address is specified with a *start* and *end* attribute. Both attributes represent the bit-address of the variable, which allows a bitwise low-level addressing of the DOM information. If the tag information should be addressed on a higher level – e.g. using a byte-wise memory access – these values have to be divided by an appropriate factor. All addresses are relative to the *offset* attribute of their parent block element. The *datatype* attribute serves as additional information for our demonstrator that can be used within the demonstrator implementation to format the variable value.

```
<block type="SmartFactory" offset="352">
 …
 <blockpart variable="NoPillA"
   datatype="DiscreteNumber"
   start="32"
   end="39"/>
 …
</block>
```

**Listing 1. XML Description of the Variable "NrPillA"**

After the variable address has been retrieved from the XML document it is sent back to the PLC which uses them to decode the variable values from the internal memory buffer. In case of the "NoPillA" variable the binary memory address "32" is sent, using a predefined communication protocol. This starting address is sufficient for the PLC to decode the variable value, since the end address of the variable can be derived from the static length of the variable data types – e.g. BOOL, BYTE and WORD.

The benefit of this approach is that the memory address can be changed in the XML document without touching the PLC code. To evaluate the concept a demonstrator has been developed which is introduced in the next section of this paper.

# 4. PROOF OF CONCEPT

The feasibility and viability of our approach has been proven by a demonstration during the Hannover Messe Industry Fair 2010. The demonstration was embedded in a scenario that used a digital object memory (DOM) on an exemplary product (a so called smart drug case) to store information relevant during various stages of the product life cycle, and share this information across the entire value chain via standardized DOM structure and access mechanisms.

The SmartFactory demonstrator (see Figure 3) constituted one of the manufacturing modules in this scenario, and used information on the DOM to control its production process of filling the smart drug case with an individual mixture of three different types of dietary supplement pills and capping the final product. Additionally, quality control information (weight) was stored in the product's memory and could be used in subsequent process steps.

The industrial hardware of the two demonstration modules (filling module and robot module) consists of programmable logic controllers (PLC), RFID read/write devices, pneumatic components, an industrial robot, sensors and actuators. The mapping mechanism is implemented using a client/server approach, with the mapping server running on an external PC and the client running on the PLC.

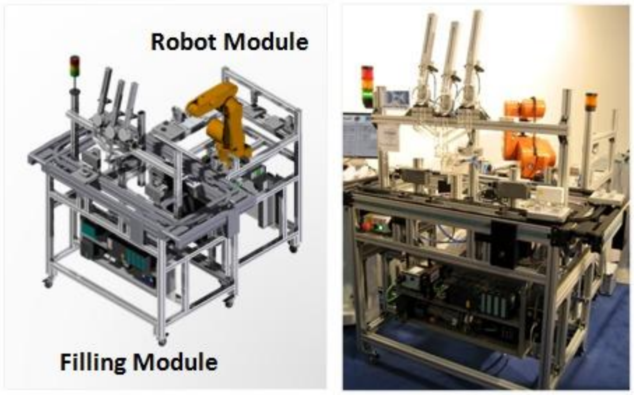


**Figure 3. Setup of the Hannover Fair Demonstrator**

The server application on the PC is implemented as a Java console application, opening a new thread for each accepted look-up request, performing the look-up of the requested variable in the appropriate XML document, sending back the result and closing the thread. The client-side application consists of a function block being called by the PLC for each read/write operation to the DOM. This function block establishes a TCP/IP connection to the mapping server sending out a variable request according to a predefined communication protocol. The response is then processed and information read from or written to the appropriate positions in the DOM. Since all these steps happen "behind the scenes" inside the demonstration module, a visualization with live data is implemented using JavaFX[1], to give visitors a better understanding of the mapping concept and mechanism (see Figure 4).

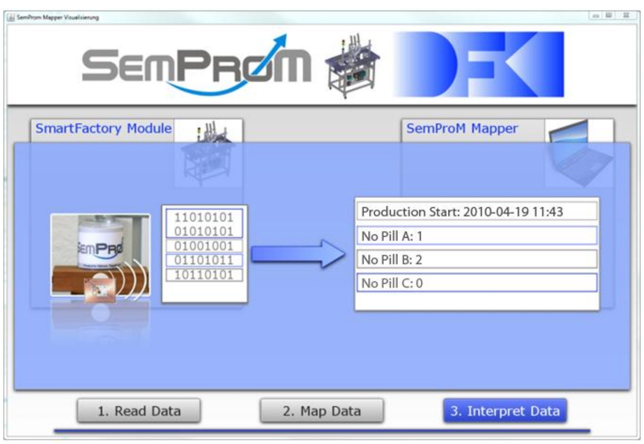


**Figure 4. Visualization of the Demonstrator**

The entire setup proved to be stable, running five days without any problems. Changes in the structure of the DOM could be ported to the whole demonstration system by simply changing the server-side XML document, eliminating the need of individual PLC reprogramming.

## 5. BENEFITS AND LIMITATIONS

In the world of decentralized industrial automation, implementing seemingly small changes in the structure of a DOM can be extremely cumbersome and complex. Switching the order of a few bytes can result in a reprogramming and re-release effort affecting a number of systems on all levels on the automation pyramid (ERP, MES, PLC). The presented approach of decoupling variables from their DOM-specific memory address shows a way to overcome these obstacles that stand in the way of a widespread, successful use of DOMs in the production industry.

The migration to new technologies on the DOM-side of the process (e.g. bigger memory, added computing intelligence) or on the RFID reader/writer and PLC side (e.g. new standards) can easily be accomplished, because the code to interpret the information on the DOM is separated from the rest of the application code and does not have to be changed.

With increasing flexibility of production plants, individualization of products and shortening product life cycles, the support for change management and version control become crucial factors influencing the adoption of new concepts like DOMs. The presented approach simplifies the introduction of new versions of DOMs. A rearranged information structure just has to be documented in XML, uploaded to the mapping server and for all DOMs carrying a version flag, the appropriate new structure definition can be loaded automatically by the PLC. Additionally, a DOM structure modeled with XML-formats can be verified against the schema definition, increasing the safety of change processes.

In production plants, each change of the PLC code has to be followed by a costly technical release procedure to ensure that process and production quality are still being met. In the proposed approach, the information about how the bits and bytes on the DOM should be interpreted is no longer a part of the PLC program but instead documented in an external file on a server. Changes to this file may not require a full technical release process, since the PLC code remains untouched. In real-life production plants, this could potentially save time and money.

Besides the outlined benefits we also identified some limitations of our approach that have to be considered in the future work:

While the approach allows a flexible change of the variable memory addresses without touching the PLC code, a change of the hierarchical information structure or the variable names in the XML document still demands for a change of the PLC code.

A further limitation is that the used client-server-architecture is less suitable for applications with hard real-time constraints. This design decision has been made because today's PLCs do not support the adequate processing of XML documents. To ease the processing of the XML document it has been outsourced to a external server application. Although restrictions concerning the real-time capability may apply, the client/server-communication is limited within our demonstration scenario to a minimum, since "unknown" DOM variables are only requested once by the PLC, which has a positive effect on the real-time capability of the system.

## 6. FUTURE RESEARCH DIRECTIONS

Digital Object Memories are a means to provide value-added services to the entire value chain. The solution presented in this paper increases the cross-domain applicability of DOMs by adding an infrastructure to exchange the underlying information models between all stakeholders of the value chain in a human- and machine-readable format.

[1] http://www.javafx.com

The separation of information and their binary representation on a DOM, as well as the annotation of the binary values with meta-information, is a prerequisite to increase the self-descriptiveness of Digital Object Memories. Consequently this helps to close the semantic gap between storage DOMs – such as RFID-tags – and applications. Although the presented approach represents a first step towards bridging the gap, the ultimate goal will be to store a data scheme on the DOM hardware itself. This would enable to use intelligent DOM readers that offer a semantic interface for the DOM access.

Having a semantic interface would allow using enriched information – instead of bit-strings – within the application, which offers the technical basis for major enhancements of industrial production systems by means of flexibility and changeability. First, the presented technology enables the ad hoc networking of DOMs with applications that are not specifically tailored to a unique hardware. This enhances flexibility and adaptivity in production systems. Second, the presented work opens the field of semantic technologies for DOMs in the area of industrial production.

Our future research will concentrate on increasing flexibility and agility of industrial production systems by using DOMs and semantic technologies in all fundamental areas of industrial production. Here, major examples are the integration and processing of context information – e.g. offered by DOMs – in industrial processes, the introduction of the paradigm of service-oriented architectures (SOA) to the lower levels of industrial automation for DOM-based invocation of functionality or the ad-hoc generation of super-ordinate control structures – such as resource-saving strategies – based on DOM information.

DOMs offering fine grained product related information using a semantic interface are an essential source of context information. The semantic processing of such context information and the situation oriented composition of data hierarchies will significantly reduce complexity in manufacturing, maintenance and logistic processes as well as in product quality assurance. One of the main challenges is to filter those context information out of the mass of available data that are provided by DOMs, sensors, tasks and roles.

Another field of application for semantic technologies in industrial automation is the semantic description of equipment functionalities as services. Using this semantic description DOMs are enabled to orchestrate the complete production process by storing the different steps that the product has to pass through during its fabrication. Combining semantic technologies, service-oriented control architectures on the lower level of industrial automation and DOM-based orchestration of production processes, the ad hoc adaptivity of a production line for a lot size of one can be achieved. This would represent an enormous improvement over today's automation technologies capabilities.

Beyond, the presented technology enables to set up super-ordinate control structures ad-hoc. Super-ordinate control structures in production processes implement higher strategies like resource optimization, long term stability of production processes or productivity optimization. The presented approach offers hardware independence of the implementation and openness of DOM information. This means that the use of standard interfaces will help to minimize integration efforts significantly. Based on this technology fine-grained product-related information on DOMs – such as energy consumption information – can offer the as-is state of production systems as well as the target state.

# 7. ACKNOWLEDGMENTS

The research described in this paper was conducted within the project Semantic Product Memory (SemProM). SemProM is funded by the German Federal Ministry of Education and Research under grant number 01 IA 08002. The responsibility for this publication lies with the authors.

The authors also would like to thank all colleagues, partners, and members of staff who contributed to this paper and the work presented.

# 8. REFERENCES

[1] Bullinger, H. and ten Hompel, M. Internet der Dinge. Selbststeuernde Objekte und selbstorganisierende Systeme, Berlin: Springer (2007).

[2] Daniel Peintner, Santiago Pericas-Geertsen: Efficient XML Interchange (EXI) Primer. W3C, 19. Dezember 2007, Accessed 7. June 2010

[3] Engels, D., Foley, J., Waldop, J., Sarma, S., and Brock, D. The Networked Physical World: An Automated Identification Architecture, *2nd IEEE Workshop on Internet Applications* (2001).

[4] Fleisch, E. and Mattern, F. Das Internet der Dinge: Ubiquitous Computing und RFID in der Praxis, Berlin: Springer (2005).

[5] Gershenfeld, N., Krikorian, R. and Cohen, D. The Internet of Things. In: Scientific American, 291, 4 (2004), 76-81.

[6] Loebbecke, C. RFID technology and applications in the retail supply chain: The early Metro Group pilot, In: Proc. of the 18th Bled eConference (2005).

[7] Mase, K., Sumi, Y., and Fels, S. Memory and Sharing of Experiences, In: Personal and Ubiquitous Computing, 11, 4 (2007).

[8] Michael, K. McCathie, L.: The pros and cons of RFID in supply chain management. In: Proc. of the International Conference on Mobile Business (2005), 623-629.

[9] Neidig, J. and Opgenoorth, B. RFID in der Automatisierung - ein Blick in die Zukunft, In: atp – Automatisierungstechnische Praxis, 7 (2008), 34-38.

[10] Neidig, J., Stephan, P.: An Object Memory Modeling Approach for Product Life Cycle Applications. Proc.of the 1st Workshop on Digital Object Memories. 5th Intl. Conf. on Intelligent Environments (IE'09) (2009).

[11] Plate, C., Basselin, N., Kröner, A., Schneider, M., Baldes, S., Dimitrova, V., and Jameson, A. Recomindation: New Functions for Augmented Memories, In: Proc. of the International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH2006), 141-150 (2006).

[12] Römer, K., Schoch, T., Mattern, F., and Dübendorfer, T. Smart Identification Frameworks for Ubiquitous Computing Applications, In: *Wireless Networks, Special issue: Pervasive computing and communications*, 10, 6 (2003), 689-700.

[13] Schneider, M. and Kröner, A. The Smart Pizza Packing: An Application of Object Memories, In: Proc. of the 4th

International Conference on Intelligent Environments (IE'08), 1-8 (2008).

[14] Schneider, M. Towards a General Object Memory, 1st International Workshop on Design and Integration Principles for Smart Objects in conjunction with UbiComp'07 (2007).

[15] Stephan, P., Meixner, G., Koessling, H., Floerchinger, F. and Ollinger, L.: Product-Mediated Communication through Digital Object Memories in Heterogeneous Value Chains. Proc. of the 8th ann. IEEE Intl. Conf. on Pervasive Computing (PerCom 2010) (2010).

[16] W3C: Extensible Markup Language (XML) 1.0 (Fifth Edition) – W3C Recommendation, http://www.w3.org/TR/REC-xml. Website, Accessed 26. November 2008.

[17] Wahlster, W., Kröner, A., Schneider, M., and Baus, J. Sharing Memories of Smart Products and Their Consumers in Instrumented Environments, In: it - Information Technology, 50, 1 (2008), 45-50.

[18] Weinländer, M. Auto-ID in der Produktionssteuerung, In: Bartneck, N., Klaas, V. and Schönherr, H. (ed.): Prozesse optimieren mit RFID und Auto-ID. Grundlagen, Problemlösung und Anwendungsbeispiele, Erlangen, Publicis Mcd (2008).

[19] Zuehlke, D. SmartFactory – From Vision to Reality in Factory Technologies, In: Proc. of the 17th International Federation of Automatic Control (IFAC) World Congress, 82-89 (2008).

[20] Zuehlke, D.: Smart Factory – Towards a factory of things. Annual Reviews in Control. Vol. 34, Issue 1, P 129-138, ELSEVIER, April 2010.