An Object Memory Modeling Approach for Product Life Cycle Applications

Joerg NEIDIG[a, 1] and Peter STEPHAN[b, 2]
[a] *Siemens AG, Germany*
[b] *German Research Center for Artificial Intelligence, Germany*

Abstract. Today, industrial production and supply chains are facing increased demands regarding flexibility and transparency of processes, caused by a trend for mass-customization and increasingly tighter regulations for the traceability of goods. To fulfill such challenging market demands, auto-ID technologies and semantic product descriptions are becoming part of future value chains. In this paper a modelling approach for a digital object memory (DOM) allowing for the attachment of product life cycle (PLC) information to everyday objects is presented. After reporting on the design aims, memory architecture and data structure, potential benefits of the chosen approach are presented.

Keywords. Digital Object Memory, Modeling, Internet of Things, Distributed Systems, Product Life Cycle, Radio Frequency Identification.

1. Introduction

Today, the increasing trend of shortening product life cycles and the customer desire for highly individualized goods are asking for flexible industrial and supply chain processes which are able to adapt quickly to changing demands. When it comes to quality products, perishable goods or healthcare products, this is supplemented by the request for a transparent monitoring of events that happened during the product's life [1]. In order to face such challenges competitively, the collection, storage and management of comprehensive product life cycle (PLC) information becomes a crucial factor for optimizing processes. In this context, auto-ID technologies like barcode and radio frequency identification (RFID) have already proven their potential to align the physical flow of goods with the digital flow of corresponding information [2].

Most of current solutions in operation use auto-ID technologies and object-related information only in the context of closed-loop applications in single domains or even a single company [3]. Although ID system solutions like the electronic product code (EPC) or the GS1 coding system already allow for a information exchange between certain stakeholders of a value chain [4, 5] these systems focus entirely on referencing information in dependence of the object's ID.

---

[1] Joerg Neidig: Siemens AG (Sector Industry), Gleiwitzerstr. 555, 90575 Nürnberg, Germany; E-mail joerg.neidig@siemens.com
[2] Peter Stephan: German Research Center for Artificial Intelligence (DFKI GmbH) Trippstadter Str. 122, 67663 Kaiserslautern, Germany; E-mail: peter.stephan@dfki.de

In this paper, the modeling approach for a digital object memory (DOM) is presented, to associate object-related PLC information with physical products in an efficient way. The solution will be tailored to the needs of real life applications, and allow for the gathering and storage of information from different sources. Furthermore the DOM will represent its content in a way enabling a seamless exchange and use of PLC information in applications throughout the whole value chain.

2. Related Work

Going conceptually beyond current auto-ID applications, DOM describe an approach to flexibly associate digital information items with physical objects [6, 7]. They are created by gathering and storing information from information sources in the environment (e.g. sensor networks) or processes a physical object participates in.

Exemplary research projects concerned with the creation of DOM are *SPECTER* [8] or *SharedLife* [9]. In *SPECTER*, a DOM is utilized to deliver ad-hoc assistance in a CD shopping scenario by triggering situation-aware mechanisms based on previously recorded user interactions. *SharedLife* drives this idea one step further by capturing, sharing and exploiting cooking experiences through DOM in a *SmartKitchen* environment. As other implementation examples show, DOM can vary in several dimensions like location of the physical data storage [3, 8], implementation approaches [10, 11] and potential application contexts [12, 13, 14].

By accompanying a smart pizza packaging through stages of its PLC, it has been proven in [15] that the concept of DOM also works for dealing with PLC information in broader application contexts like a complete value chain.

Numerous standards for object descriptions have been created, all aiming at the definition of a comprehensive list of object properties. Well renowned approaches are the *Electronic Device Description Language* (EDDL) [16], the *Field Device Tool* (FDT) [17] the *Physical Markup Language* (PML) [18] or the *Smart Description Object* (SPDO) [19]. As all these models are created for a certain application or domain, they are deemed not be abstract and generic enough to describe all kinds of objects throughout an entire PLC.

3. Object Memory Design

3.1. Design Aims

The aim regarding the DOM design presented is not to simply enrich the world with another model for describing a set of object-related information. Instead, the idea is to create a container format for object-related information which does not substitute the different device and object description languages, but unites them under one roof. The requirements for the approach are given below.

1. The DOM must contain a number of mandatory entries (e.g. object ID)
2. A set of information has to be intelligible to all stakeholders of the value chain, i.e. can be accessed and understood throughout the whole PLC
3. Besides common DOM information, an option to embed information given in proprietary formatting must be offered, in order to enable stakeholders to include already existing data structures

4. The DOM has to protect itself, meaning that access should only be possible via an interface, which also allows to search for specific content
5. To allow for a hardware independent DOM implementation, the model needs sufficient flexibility and scalability
6. The DOM has the option to link to external information sources, i.e. incorporates arbitrary information entities like single parameters, entire information blocks or even the complete DOM
7. The DOM has to be identifiable and allow for the versioning of it's states

3.2. DOM Architecture

The key to enable a DOM solution for dealing with information over a whole PLC lies in the flexibility of its architecture. To allow for a maximum of adaptability and openness, all DOM information is arranged in a modular block structure similar to a file system (see **Figure 1**). The structure consists of distinct blocks each with a clear function and determined content.

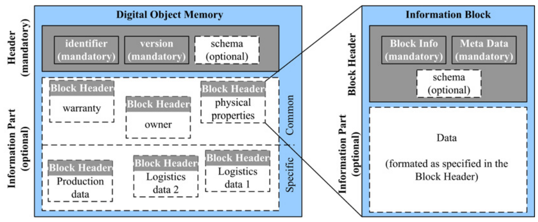

Figure 1: Structure of the DOM and its information blocks.

In the presented architecture, the *header* is a mandatory part of the DOM including all information needed to identify this model as a valid DOM of a certain version. Optionally, the header can be extended to include a schema for the DOM model making it self-explanatory. The *information part* contains all the information related to the object itself. It consists of a collection of blocks filled with different types of information. These blocks can be coarsely divided into common and specific ones. The *common* blocks contain information which is not specific to a certain application, domain or section of the PLC, e.g. physical object properties, warranty information or hazard statements. To make this information commonly understandable the available keywords and the structure are fixed. The *specific* blocks contain information that is only of interest for a limited set of users. As this information is likely to be proprietary (e.g. production parameters or sensor readings) only very little restrictions are made, allowing to embed almost any kind of information.

The modular structure of the information part offers high flexibility and scalability. The combination of a defined structure, which gives the DOM a unified appearance and the support for any kind of information formats make the DOM a convenient information pool. The chosen design allows for the retrieval of specific information by simply accessing specific blocks. Furthermore, with the presented structure the DOM can be extended easily by simply adding further blocks to the information part.

3.3. Information Block Description

The structure of an information block is simple and independent of it's content. It consists of a *Block Info*, *Metadata*, and the *Data* itself. Whereas the Block Info and

Meta Data are mandatory using fixed keywords, the Data may be given in any format desired.

The *Block Info* is a short section which contains the block characteristics *Block Type*, *Block Format* and its *ID*. By that, the Block Info includes the information needed to access and parse the block's *Data* section. The *Block Type* categorizes the block itself. If it is a common information block the type must be taken from the list given in the DOM specifications (e.g. "owner", "physical properties", …). If the block is a specific one the Block Type is set to "none". The *Block Format* specifies the block's data format given as a mime-type. In all cases (with the exception of "none") an XML-stream is expected in the Data section. The *ID* is unique to the block and is the key for accessing it's information content.

The *Meta Data* contain all information to classify and label the content of the Data section. It contains tags like the block's name, search keywords, the block's creator, a history log listing the changes made to the block, and so on. Consider the Meta Data section as the source of information needed for powerful search applications.

The *Data* section contains the object-related information itself given in the format specified by the Block Info. In this section any kind of information can be stored without restrictions, but it is expected that the Meta Data are kept accurate. Optionally, it is possible to store the schema of the data format in the block header. It is important to keep in mind that for the user of a DOM it is of no interest how any of the data mentioned above is stored physically. Common understanding is that on RFID-transponder or embedded devices a highly compressed bit-coding is needed whereas on pc-based devices a more elaborate encoding might be possible.

3.4. Interface Description

An application utilizing DOM information must not access memory content directly, but use a standardized interface instead (see **Figure 2**). In that context, the interface assumes several tasks. It works as an abstraction layer from the DOM hardware by translating the data from the format in which it is stored physically (e.g. bitcode on an RFID-tag) into the format expected by the user (e.g. a XML-stream) and vice versa. Furthermore, it has to act as a kind of administration software by creating, organizing and deleting DOM blocks and implementing rudimentary search capabilities. Information from a DOM is accessed via the ID of the respective information block, e.g. the data of Block 42 is accessed via the function `GetUserData(42)`. As the IDs are generated by the interface automatically, they have to be retrieved via a search operation implemented in the interface. For example, the command `GetBlockIds()` will return the IDs of all blocks, `GetBlockIds(`*Metadata*`)` returns the IDs of blocks with certain Meta Data. More complex searches operations have to be implemented in external applications. The intended way to access the DOM is to retrieve a list of relevant IDs through the search option, then to (partially) retrieve the respective blocks and finally to filter the information in the application. This has the advantage of keeping the interface layer lightweight by limiting its complexity. The creation of a new block is initialized via the function `CreateBlock()` returning the ID of the new block. User and Meta Data can be added or altered with the appropriate `Set` functions.

Figure 2: Object Memory Interface.

4. Benefits

Section 2 showed that a large number of different object description languages and data formats specific to certain domains and optimized for limited phases of the PLC exist. The presented approach offers a container in which existing ways to describe object-related information can be united. This will allow for the gathering and storage of information over the whole PLC always in the most appropriate way. By that, potential users can stick with the object languages and data formats already in use. In order to migrate to the DOM framework, simply an interface plug-in is required. Regarding the presented DOM container itself, its flexible architecture sets it apart from other current approaches. Adopting a structure consisting of distinct information blocks allows to interact with small subsets of DOM information. i.e. initializing, searching and parsing DOM content can be limited to only relevant blocks without loading the DOM completely. By that, only little computing power and small communication bandwidth are sufficient to interact with the DOM, enabling it for lightweight smart item solutions attached directly to a product.

To verify the viability of the presented approach, an initial DOM version has been implemented in a hardware demonstrator at this year's CeBIT 2009 fair. Experiences and further benefits are reported in a publication currently under review [20].

5. Conclusion

In this paper a modelling approach for a digital object memory (DOM) is presented, allowing for the effective attachment of product life cycle (PLC) information to everyday objects. Based on a number of design aims, the development of a container format is described, in which existing ways to describe object-related information can be united. The key to enable such functionality lies in the flexible DOM architecture and the information storage in a block structure which allows for a maximum of openness in information handling. The access to the DOM is managed by a standardized interface. Main benefits of the presented approach are its capabilities to integrate object-related information of different domains over a whole PLC, it's easy expandability and the possibility to interact only with DOM information blocks of interest.

Acknowledgements

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.

References

[1] Garcia, A., McFarlane, D., Fletcher, M., and Thome, A. Auto-ID in Materials Handling, Cambridge, MA: White paper, Auto-ID Center, MIT (2003).

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

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

[4] 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).

[5] 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.

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

[7] Sumi, Y., Sakamoto, R., Nakao, K., and Mase, K. ComicDiary: Representing individual experiences in a comis style, In: *Proc. of the 4th International Conference on Ubiquitous Computing* (2002).

[8] 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).

[9] Schneider, M. The Semantic Cookbook: Sharing Cooking Experiences in the Smart Kitchen, In: *Proc. of the 3rd International Conference on Intelligent Environments (IE'07)*, 416-423 (2007).

[10] 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.

[11] Barbu, C., and Kröner, A. Designing a Study Concerning the Functions of Sharable Personal Memories, In: *Proc. of the IADIS International Conference WWW/Internet 2008* (2008).

[12] Pollack, M. E. Intelligent technology for an aging population: The use of AI to assist elders with cognitive impairment, In: *AI Magazine*, 26, 2 (2005), 9-24.

[13] Kuwahara, N., Noma, H., Kogure, N., Tetsutani, N., and Iseki, H. Wearable auto-event-recording of medical nursing, In: *Proc. of the 9th IFIP TC13 International Conference on Human-Computer Interaction (INTERACT'03)*, (2003).

[14] Wasinger, R., and Wahlster, W. The antropomorphized product shelf: Symmetric multimodal interaction with instrumented environments, In: Aarts, E., and Encarnacao, J. (ed.): *True Visions: The Emergence of Ambient Intelligence*, Heidelberg, Berlin, New York, Springer (2006).

[15] 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).

[16] Riedl, M., Simon, R., Thron, M. *EDDL – Electronic Device Description Language*, Oldenbourg, (2001)

[17] Simon, R., et al. *FDT Field Device Tool*, Oldenbourg, (2005)

[18] Brock, D.L., et al. *The Physical Markup Language*, White Paper, MIT Auto-ID Cener, (2001)

[19] Janzen, S. and Maass, W. Smart Product Description Object (SPDO), In: *Poster Proc. of the 5th International Conference on Formal Ontology in Information Systems (FOIS2008),* (2008)

[20] Stephan, P., et al. Product Mediated Communication Trough Digital Object Memories in Heterogeneous Value Chains, Under Review: *11th Intl. Conference on Ubiqitous Computing,* (2009).