Ontology Engineering for Collaborative Embedded Systems – Requirements and Initial Approach

C. Hildebrandt [1], S. Törsleff[1], T. Bandyszak[2], B. Caesar[1], A. Ludewig[1], A. Fay[1]

Abstract: Collaborative embedded systems (CES) heavily rely on information models to understand the contextual situations they are exposed to. These information models serve different purposes. First, during development time it is necessary to model the context for eliciting and documenting the requirements that a CES is supposed to achieve. Second, information models provide information to simulate different contextual situations and CES´s behavior in these situations. Finally, CESs need information models about their context during runtime in order to react to different contextual situations and exchange context information with other CESs. Heavyweight ontologies, based on Ontology Web Language (OWL), have already proven suitable for representing knowledge about contextual situations during runtime. Furthermore, lightweight ontologies (e.g. class diagrams) have proven their practicality for creating domain specific languages for requirements documentation. However, building an ontology (light- or heavyweight) is a non-trivial task that needs to be integrated into development methods for CESs such that it serves the above stated purposes in a seamless way. This paper introduces the requirements for the building of ontologies and proposes a method that is integrated into the engineering of CESs.

Keywords: Collaborative Embedded Systems, Ontology Building, Requirements

1 Introduction

Embedded systems are deeply integrated into their context, since they have to sense contextual situations by means of sensors and interact with objects in the context by their actuators. However, due to the rising complexity of the tasks assigned to embedded systems, modern embedded systems are increasingly interconnected, and collaborate with other systems to achieve common goals that a single system could not achieve on its own. In this paper, we refer to such systems as collaborative embedded systems (CESs). For example, in the manufacturing domain, several application scenarios of the platform “Industrie 4.0” [PLA16@] illustrate the collaboration of embedded systems. In the application scenario “Order Controlled Production” for instance, CESs are supposed to autonomously manufacture a customer order from inquiry to finished product. In the energy domain, CES are of growing importance as well, as decentralized control solutions based on distributed intelligent components are considered a key enabler for the “Energiewende” [De12@]. As a result, much research in the fields of distribution grid automation and virtual power plants is concerned with the utilization of CESs that collaborate with one another to achieve common goals. In both examples, the CES are

[1] Helmut-Schmidt-University, Institute for Automation Technology, Holstenhofweg 85, 22043 Hamburg, {c.hildebrandt, toersles, caesarb, ludewiga, a.fay}@hsu-hh.de

[2] University of Duisburg-Essen, paluno – The Ruhr Institute for Software Technology, Gerlingstraße 16, 45127 Essen, Torsten.Bandyszak@paluno.uni-due.de

required to exchange complex information about the context they are embedded in. Furthermore, they need a common understanding regarding the syntax and semantic that is being exchanged. Hence, there is an imminent need for information models that allow for modelling the context of different CESs in a unified manner. Underlying each information model is an ontology, i.e., a theory about the real-world objects to be modelled [STW03]. According to [JU99], an "ontology may take a variety of forms, but it will necessarily include a **vocabulary of terms** and some **specification of their meaning**. This includes definitions and an indication of **how concepts are inter-related** which collectively impose a structure on the domain and constrain the possible interpretations of terms." Ontologies have already proven a promising candidate for modeling complex contextual relations in the domain of context-aware applications [BBH+10]. Furthermore, there is evidence in the manufacturing domain that proves the applicability of ontologies for modeling complex information [NFG+16]. However, there is no universal and seamless method for developing ontologies that address the specific characteristics of complex information models in the engineering of CESs and the different domains that CESs are applied to. The contribution of this paper is threefold: First, we analyzed industry requirements elicited within the research project CrESt[2] (Collaborative Embedded Systems), in order to identify different applications of ontologies in the engineering of CESs, and to extract requirements for the development of respective ontologies (Section 2). Second, a first approach towards a seamless method for ontology building in the engineering of CES is proposed (Section 3). Third, the first methodological building block of this method, the ontology requirements specification (ORS), is discussed in detail (Section 4). Section 5 contains a summary and an outlook to conclude the paper.

2 Requirements and Related Work regarding Ontologies in the Development of Collaborative Embedded Systems

This section presents the results of an analysis of industry requirements on information models and methods for their development, as well as desired applications of information models in the context of CES. Furthermore, requirements concerning methods for ontology building that have been extracted from industry requirements are presented.

Industrial Requirements and Applications

In CrESt, a total[3] of 22 requirements regarding models and 16 requirements regarding methods have been collected by industry partners with respect to the context of CES in the Use Cases "Flexible and Adaptable Factory (UCF) and "Distributed Energy Production" (UCE). In the first step, these requirements regarding the models have been analyzed in comparison in order to identify commonalities. The result of this analysis are keywords that have been used for creating a classification of requirements. We identified

[2] See https://crest.in.tum.de

[3] Not including sub-requirements

three categories that are labeled with the key words: **(1)** “Semi-formal” or “meta model”; **(2)** “Simulation” or “Executable”; **(3)** “Machine readable” or “data model”. The results are shown in Table 1. These key words cover all requirements for UCF and 75% of the requirements of UCE. The extracted common requirements, represented by the three categories, can therefore be assumed to be valid for both domains. Even though the content of the underlying requirements in the requirement classes is heterogeneous, there is a common need for information models in the requirements, which can be addressed by an ontology.

Table 1: Key-word analysis of requirements in the two use cases

Use Case	“Semi-formal” model	“Simulation” or “Executable” model	“Machine readable” or “data” model
Distributed Energy (UCE)	2 out of 4	-	1 out of 4
Adaptable Factory (UCF)	4 out of 18	10 out of 18	4 out of 18

In the second step, the commonalities have been analyzed to address the common needs. This analysis of the requirement categories has shown three possible applications of ontologies: **(A1)** Ontologies for the creation of profiles that can be used for the semi-formal modelling of context. In this application of an ontology, an existing meta model of the Object Management Group is extended by using a domain specific vocabulary that is modelled with UML (Unified Modelling Language) class diagrams. A process description for extending the meta model is described in [TBW13] and [FV04]. **(A2)** Ontologies for being used to represent contextual situations for the purpose of simulating the behaviour of a CES in different contextual situations. The importance of knowledge representation has already been identified in the in the simulation community. In [TU14] for instance, ontologies are used for the description of properties of a factory’s objects, which can be used for simulation of process plans. In [DÖ16], an ontology for representing artefacts in simulation systems engineering is proposed. **(A3)** Ontologies for the purpose of information provision at run-time for, or between, context-aware applications like scheduling of manufacturing operations under changing contextual conditions. Applications using ontologies for this purpose can be found in [WDY+17]. In [AVH16], a context-aware maintenance system is proposed that uses an ontology for representing the shared knowledge between systems and in [PLM13] and [HLC14] ontologies are used for providing the necessary information for orchestrating manufacturing services. However, the requirements that ontologies have to fulfil in these different applications (A1 – A3) are not identical. In Ontology Engineering one can distinguish between lightweight (e.g. UML class diagrams) and heavyweight (e.g. OWL Description Logics) ontologies [GFC04]. Applications such as (A1) require lightweight ontologies, whereas applications such as (A2) and (A3) call for heavyweight ontologies that provide a higher degree of formalism than lightweight ontologies do. Regarding the methods concerning the modelling of context, 5 out of 12 (UCF) and 3 out of 4 (UCE) requirements concern the creation of models, which leads to the building of aforementioned ontologies. Due to the different applications (A1-A3) of ontologies in the field of CESs and also the amount of different models required by industry, a universal method for building ontologies in the domain of CES is to be preferred at the moment, instead of ad-hoc building application-specific ontologies for both, scientific community and industry partners. To summarize, two statements can be made based on the aforementioned analysis. First, ontologies are necessary for solving the upcoming challenges in designing CESs since they at least help addressing most model-related requirements. Second, a method is needed for the creation of these ontologies that can be applied in the development process of CES. Even though several approaches with respect to ontology building have been proposed, e.g. METHONTOLOGY [FGJ97], DILIGENT [PST04], On-To-Knowledge [SSS+01], NeOn Methodology [SGM+12] and UPON [NMN09], none of them fulfils all of the requirements regarding ontology building, which are listed below. Therefore, a method has been developed that addresses the requirements regarding context modelling and the related methods; this method is briefly introduced in section 3.

Requirements regarding Ontology Building

Additional requirements with respect to ontology building have been extracted from the industry requirements and are discussed subsequently. The method presented in section 3 has to fulfil the following, ontology building-specific requirements: **(R1) Iterative procedure**: Information models in the engineering domain can become very complex. The Standard for the Exchange of Product Model Data (STEP) can be taken as an example here. The STEP standard consists of 55 sub standards [ISO 10303-1] that define how the data of a product during its life cycle can be modelled. For the purpose of bringing this information into an ontological structure, an iterative procedure is necessary such that the resulting ontology caters the actual needs of an engineering project. **(R2) Modularization of knowledge**: Since the information in the different domains of CESs is very complex, the methodology should explicitly support the modularization of the domain´s knowledge in order to support the reuse of existing ontologies and also allow for application-specific parts of the ontology instead of providing the full amount of knowledge in one model. **(R3) Different solution paths**: The different addressed applications have different starting points for ontology creation. For some of the information necessary in the UCF for instance, ontologies are readily available (e.g. ontology design patterns[4] or complete ontologies). On the other hand, an ontology that specifically addresses the context of collaborative embedded systems would have to be created from scratch. **(R4) Applicable for lightweight and heavyweight ontology building**: For the development of CESs there are different levels of formalism necessary. In order to create a profile for instance, the formalism necessary can be modelled with a class diagram. For applications involving simulations, however, logically interpretable ontologies are required, for instance description logics. Thus, the method shall support different levels of formalism. **(R5) Seamless creation of light and heavyweight ontologies**: Since a lightweight ontology already contains some of the information necessary for a heavyweight ontology, the method should support the

[4] For instance: http://ontologydesignpatterns.org/wiki/Submissions:MaterialsProperty

seamless development of ontologies from ontological requirements via lightweight ontologies to heavyweight ontologies. **(R6) Reuse of existing ontologies**: the reuse of ontologies should be supported in a structured manner. **(R7) Reuse of non-ontological resources**: For most information to be modelled, there already exists an enormous amount of information resources (e.g. national & international standards, company-specific process description, classification systems like ecl@ss etc.). The method for the creation of an ontology has to allow the structured use of such information pools. **(R8) Integration into the engineering workflow**: Since ontology building is a process that will be performed in addition to the engineering of CES, it is necessary that the interdependencies with the engineering workflow are explicitly considered.

3 An Initial Method for Ontology Building for Collaborative Embedded Systems

Based on the elicited industry requirements and the discussion in section 2, a method has been developed that addresses the discussed requirements. The method is divided into multiple MBBs as it has been proposed in [DBB+16]. Note, that a detailed explanation of every building block is out of scope of this paper (see section 5) and instead subject of future contributions. The method is shown in Fig. 1, while the satisfaction of R1-R8 is subsequently discussed. The method distinguishes between two types of artefacts: pre-existing artefacts that can be reused in engineering projects (shown in blue in Fig. 1) and

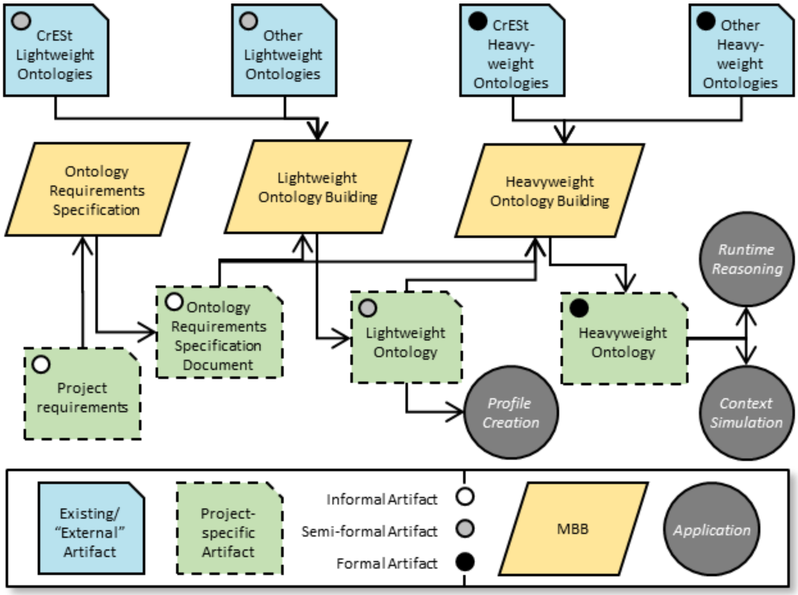

CrESt Lightweight Ontologies
Other Lightweight Ontologies
CrESt Heavy-weight Ontologies
Other Heavy-weight Ontologies
Ontology Requirements Specification
Lightweight Ontology Building
Heavyweight Ontology Building
Runtime Reasoning
Ontology Requirements Specification Document
Lightweight Ontology
Heavyweight Ontology
Project requirements
Profile Creation
Context Simulation
Existing/ "External" Artifact
Project-specific Artifact
Informal Artifact
Semi-formal Artifact
Formal Artifact
MBB
Application

Fig. 1: Initial method for Ontology Building for Collaborative Embedded Systems

project-specific artefacts (shown in green in Fig. 1). Furthermore, we distinguish between informal, semi-formal and formal artefacts. The ontological building process consists of MBBs that can be used, if required by project requirements, and consist of tasks that can be performed iteratively (see also section 4). In the first MBB, the ontology requirements are extracted from the project requirements. The second MBB uses these ontological requirements to build a lightweight ontology that can be used to create domain-specific profiles in order to extend existing metamodels or profiles [TBW13]. Such semi-formal context models (see [DTW12] for a detailed introduction to context models) can be modelled using UML or SysML. This MBB also considers existing artefacts of ontological and non-ontological nature. The third MBB uses the project-specific artefacts created so far (at least the ORS Document) and builds a formal ontology that can be used to formalize the semi-formal context models, which allows reasoning during runtime as well as simulation based on formal context models. The use of MBBs satisfies the requirements R2, R3 and R8. The requirements R6 and R7 are satisfied by explicit consideration of external resources. The requirements R4 and R5 are also considered by using directly related MBB for R4 as well as by the definition of an interface to lightweight ontologies in heavyweight ontology building (R5).

4 Ontology Requirements Specification for Collaborative Embedded Systems

In this section, the first MBB of the method proposed in section 3 is described. For the purpose of (ORS), the NeOn Methodology is chosen, since this method, compared to all above mentioned approaches, covers most of the requirements stated in section 2. The requirements R4, R5 and R8 are not covered, neither by the NeOn Methodology nor by any other approach listed above. This can be traced back to the different scope of the approaches which focus on heavyweight ontology development in the Semantic Web domain. The process of ORS according to [SGV09], which is part of the NeOn methodology, is shown in Fig. 2. The remainder of this section applies this approach to the domain of CES. The ORS process consists of eight tasks (see Fig. 2). However, a focus is set on tasks 1, 3 and 4, since they differ in the domain of CES from the Semantic Web domain and the method in [SGV09]. The example use case in the following is the creation of a profile according to [TBW13] for the purpose of defining a domain specific language for UCF. The documentation of the ORS can be done in a tabular format, while a template, can be accessed online[5]. The stakeholders involved in the ORS are domain experts, ontology specialists and users of the ontology. Most ORS tasks are performed jointly by all stakeholders, except for task 6 (only users and domain experts) and task 8 (only ontology specialists). The input of the ORS is a set of ontological needs. The output is a document stating all requirements that the ontology has to fulfil and that the ontology is verified against after it has been developed. In task 1, the objective is to define the purpose, scope and implementation language of the ontology. To this end, the

[5] Online link to the ORS document template: https://www.hsu-hh.de/aut/forschung/forschungsthemen/crest-modellbasierte-entwicklung-kollaborativer-eingebetteter-systeme

ontology specialists interview the domain experts and users. In case the goal is the creation of a profile, the interview would yield that the ontology purpose is the definition of stereotypes (classes and relations), tagged values and constraints (see [FV04]) that can be used for implementing a profile. The scope addresses granularity and domain coverage of the ontology. In the context of CES, the scope might refer to upper ontologies or domain ontologies. While upper ontologies consist of concepts that are abstract and address different domains, a domain ontology is limited to concepts that are specific to a certain domain, e.g. UCF [AZW06].

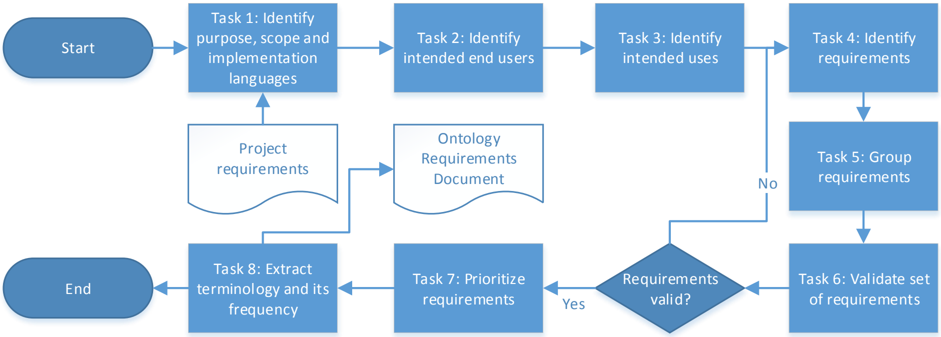
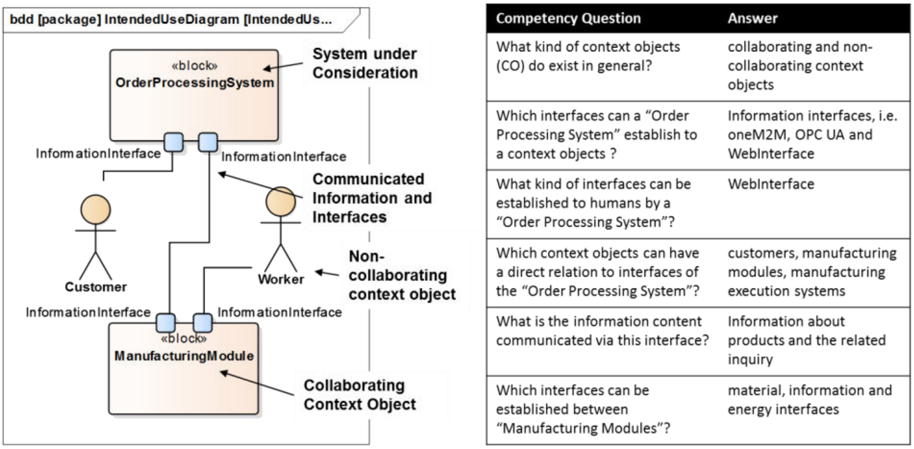

Start
Task 1: Identify purpose, scope and implementation languages
Task 2: Identify intended end users
Task 3: Identify intended uses
Task 4: Identify requirements
Project requirements
Ontology Requirements Document
No
Task 5: Group requirements
End
Task 8: Extract terminology and its frequency
Task 7: Prioritize requirements
Yes
Requirements valid?
Task 6: Validate set of requirements

Fig. 2: Tasks to be performed for ORS, based on [SGV09]

An exemplary upper ontology would be an ontology that comprises concepts and relations for UCF and UCE in a generic CES ontology. Domain ontologies on the other hand can consist of even more specific subdomains. In a domain ontology for the UCF for instance, one may define “mechanical engineering” as a more detailed domain (i.e. subdomain), or even “mechanical cylinder construction” as a subdomain of “mechanical engineering”. In our running example, the domain is defined as “Manufacturing Order Processing”. Suitable implementation languages have to be identified with respect to the ontology purpose. In the case of profile creation, SysML can be chosen in order to represent the stereotypes and relations, while OCL (Object Constraint Language) is chosen for the definition of the constraints. In task 2, the intended end users of the ontology are identified. For the running example, two intended user roles can be identified, i.e. a model based systems (MBS) engineer who performs the profile creation and a MBS engineer that uses the profile. Task 3 is concerned with specifying the intended uses of the formerly identified users. This is achieved by the definition of scenarios in which the ontology has to provide information for the purposes defined in task 1. Which procedure to use for information gathering is dependent on whether the information regarding the intended use is implicit or explicit (e.g. any pre-existing document specifying the intended uses is available) knowledge of a user or domain expert. In case the information is explicit, interviews are appropriate to gather the required information sources. In contrast, if the information is implicit, a workshop employing creative techniques (e.g. gallery method or method 635 [PBB+07]) should be held in order to identify as much scenarios as possible. A model-based documentation of the intended uses is recommended, since it facilitates the subsequent tasks. For the running example it is assumed that the users were not able to write down the intended uses or refer to any document stating the intended use, as they are not yet aware of how the target SysML profile extension will be applied in the end. Therefore, the intended use can be seen as implicit knowledge and a workshop with creative techniques has to be performed. In this workshop, the users and domain experts create SysML diagrams of possible application scenarios of the profile in small working groups. An example diagram is shown on the left side of Fig. 3, where the SysML block definition diagram is used without the profile to be developed in order to address possible applications of the profile. As a result, model elements that require the definition of ontological concepts are highlighted in the diagram. These potential applications are called “Point of Interest” (POI). For every POI in a diagram the stakeholders will define requirements in task 4. These POIs are the definition of the intended use and the result of task 3.


bdd [package] IntendedUseDiagram [IntendedUs...
«block»
OrderProcessingSystem
System under Consideration
InformationInterface
InformationInterface
Communicated Information and Interfaces
Customer
Worker
Non-collaborating context object
InformationInterface
InformationInterface
«block»
ManufacturingModule
Collaborating Context Object

Competency Question	Answer
What kind of context objects (CO) do exist in general?	collaborating and non-collaborating context objects
Which interfaces can a “Order Processing System” establish to a context objects ?	Information interfaces, i.e. oneM2M, OPC UA and WebInterface
What kind of interfaces can be established to humans by a “Order Processing System”?	WebInterface
Which context objects can have a direct relation to interfaces of the “Order Processing System”?	customers, manufacturing modules, manufacturing execution systems
What is the information content communicated via this interface?	Information about products and the related inquiry
Which interfaces can be established between “Manufacturing Modules”?	material, information and energy interfaces

Fig. 3: Example results of task 3 and task 4

After having defined the intended uses, in task 4 the technical and constraining requirements are extracted from the results of the former tasks. Technical requirements are formulated as so called competency questions (CQs), which the ontology has to provide answers to, i.e. knowledge that the ontology has to cover. These CQs can be extracted from the results of task 3. This is achieved through the definition of a CQ for each POI, which we exemplify on the right side of Fig. 3. In case the domain experts and users can already provide answers to the questions, these are documented as well. In contrast, CQs that cannot be answered due to implicit knowledge of experts and users, or due to missing information, are tagged in the ORS document to highlight the need for further investigation. The CQs and their corresponding answers are the essential requirements used later on within the process of building a light- or heavyweight ontology in order to verify whether the ontology captures all necessary concepts. In addition to the CQs, constraining requirements include general aspects like “Consideration of standard ISO 10303 for product models” or characteristics like multilingualism. In task 5, the gathered CQs and answers are grouped according to a set of different criteria by domain experts, users and the ontology specialists. These groups can be content-specific, like used measures or referenced objects (to be defined by users and domain experts) or scope-related. The scope related groups should comprise “UpperOntology”, “DomainOntology” and “SubDomainOntology”. In task 6, the users and domain experts perform a validation of the gathered requirements using a checklist (see online template). This checklist contains questions that the domain experts and users have to answer positively, in order for the requirements to be valid. This aims to ensure that the requirements are complete, consistent, verifiable, understandable, unambiguous, concise and modifiable. Upon validation, the requirements are prioritized in task 7. This is especially important if the set of requirements is large or some requirements’ implementation has to precede others. This task is performed by interviews in which the ontology specialists interview the domain experts and users. Task 8 concerns the extraction of terms and their frequency from the CQs and their respective answers in order to build a pre-glossary of terms. This pre-glossary is divided into three parts. The first part identifies terms and frequency based on the CQs, the second one is based on the related answers, and the third part identifies objects that can be seen as instances of other classes.

5 Summary and Outlook

This paper has shown the need for a seamless method for ontology building in the domain of Collaborative Embedded Systems (CES) by collecting requirements on information models proposed by industry, and by introducing possible applications of those information models as required by industry. Furthermore, an initial method has been proposed, which aims to fulfil these requirements by defining a generic process that distinguishes lightweight and heavyweight ontologies, and accounts for different artefacts as well as the potential applications of ontologies. Furthermore, we introduced the first MBB of this method. Our future research will focus on light- and heavyweight ontology building for CES to address the stated requirements. Furthermore, we will analyse the application of ontologies to CES relevant topics, e.g. system variability and open contexts.

6 References

[AVH16] Aarnio, P., Vyatkin, V., Hästbacka, D.: Context Modeling with Situation Rules for Industrial Maintenance. IEEE International Conference on Emerging Technologies and Factory Automation, 2016.

[AZW06] Aßmann, U., Zschaler, S., Wagner, G.: Ontologies, Meta-models, and the Model-Driven Paradigm. In: Ontologies for Software Engineering and Software Technology: Springer Berlin Heidelberg, 2006, S. 249–273.

[BBH+10] Bettini, C. et al.: A survey of context modelling and reasoning techniques. Pervasive and Mobile Computing, Vol. 6 (2), 2010, S. 161–180.

[DBB+16] Daun, M. et al.: Structured Model-based Engineering of Long-living Embedded Systems: The SPES Methodological Building Blocks Framework. Softwaretechnik-Trends, Vol. 36 (1), 2016.

[De12@] Deutsche Energie-Agentur: *dena-Verteilnetzstudie – Ausbau- und Innovationsbedarf der*

Stromverteilnetze in Deutschland bis 2030. URL: https://shop.dena.de/fileadmin/denashop/media/Downloads_Dateien/esd/9100_dena-Verteilnetzstudie_Abschlussbericht.pdf (last downloaded 2017-11-13), 2012

[DTW12] Daun, M., Tenbergen, B., Weyer, T.: Requirements Viewpoint. In: Model-Based Engineering of Embedded Systems: Springer Berlin Heidelberg. Berlin, Heidelberg, 2012, S. 51–68.

[DÖ16] Durak, U., Ören, T.: Towards an Ontology for Simulation Systems Engineering. In: Annual Simulation Symposium: Pasadena, California, USA, Curran Associates Inc (Simulation series, volume 48, no. 2). Red Hook, NY, 2016.

[FGJ97] Fernández-López, M., Gómez-Pérez, A., Juristo, N.: METHONTOLOGY: From Ontological Art Towards Ontological Engineering. In Proceedings of the Ontological Engineering AAAI-97 Spring Symposium Series, 1997.

[FV04] Fuentes-Fernández, L., Vallecillo-Moreno, A.: An Introduction to UML Profiles. UML and Model Engineering (2), 2004.

[GFC04] Gómez-Pérez, A., Fernández-López, M., Corcho, O.: Ontological Engineering. London: Springer-Verlag, 2004.

[HLC14] Han, S.N., Lee, G.M., Crespi, N.: Semantic Context-Aware Service Composition for Building Automation System. IEEE Transactions on Industrial Informatics, Vol. 10 (1), 2014, S. 752–761.

[ISO 10303-1] Industrial automation systems and integration - Product data representation and exchange - Part 1: Overview and fundamental principles, 15.12.1994.

[JU99] Jasper, R., Uschold, M.: A framework for understanding and classifying ontology applications. In: Int. Workshop on Knowledge Acquisition, Modelling, and Management KAW, 1999.

[NFG+16] Negri, E. et al.: Requirements and languages for the semantic representation of manufacturing systems. Computers in Industry, Vol. 81, 2016, S. 55–66.

[NMN09] Nicola, A. de, Missikoff, M., Navigli, R.: A software engineering approach to ontology building. Information Systems, Vol. 34 (2), 2009, S. 258–275.

[PBB+07] Pahl, G. et al.: Engineering Design: A Systematic Approach. Third Edition. Springer-Verlag, London, 2007.

[PST04] Pinto, S., Staab, S., Tempich, C.: DILIGENT: Towards a fine-grained methodology for DIstributed, Loosely-controlled and evolvInG Engineering of oNTologies. In: European Conference on Artificial Intelligence, Valencia, Spain, 2004.

[Pla16@] Plattform Industrie 4.0: Aspects of the Research Roadmap. http://www.plattform-i40.de/I40/Redaktion/EN/Downloads/Publikation/aspects-of-the-research-roadmap.pdf?__blob=publicationFile&v=7 [last downloaded: 2017-11-09]

[PLM13] Puttonen, J., Lobov, A., Martinez Lastra: Semantics-Based Composition of Factory Automation Processes Encapsulated by Web Services. IEEE Transactions on Industrial Informatics, Vol. 9 (4), 2013, S. 2349–2359.

[SSS+01] Staab, S. et al.: Knowledge processes and ontologies. IEEE Intelligent Systems, Vol. 16 (1), 2001, S. 26–34.

[SGM+12] Suárez-Figueroa, M.C. et al.: Ontology Engineering in a Networked World. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

[SGV09] Suárez-Figueroa, M.C., Gómez-Pérez, A., Villazón-Terrazas, B.: How to Write and Use the Ontology Requirements Specification Document. In: On the Move to Meaningful Internet Systems, Springer Berlin Heidelberg, 2009, S. 966–982.

[STW03] Shanks, G., Tansley, E., Weber, E.: Using Ontology to Validate Conceptual Models. In: Communications of the ACM, Vol. 46, No. 10, S. 85-89. ACM, 2003.

[TBW13] Tenbergen, B., Bohn, P., Weyer, T.: Ein strukturierter Ansatz zur Ableitung methodenspezifischer UML/SysML-Profile am Beispiel des SPES 2020 Requirements Viewpoints. In Software Engineering (Workshops), 2013.

[TU14] Terkaj, W., Urgo, M.: Ontology-based Modeling of Production Systems for Design and Performance Evaluation. In: Proceedings of 12th IEEE International Conference on Industrial Informatics (INDIN): Brazil, 2014.

[WDY+17] Willner, A. et al.: Semantic Communication between Components for Smart Factories based on oneM2M. In 22nd IEEE Emerging Technology and Factory Automation (ETFA), 2017.