at – Automatisierungstechnik 2019; 67(6): 504–517 **DE GRUYTER** OLDENBOURG

Survey

Felix Ocker*, Christiaan J. J. Paredis and Birgit Vogel-Heuser

Applying knowledge bases to make factories smarter

Anwendung von Wissensbasen zur Steigerung der Intelligenz von Fabriken

https://doi.org/10.1515/auto-2018-0138
Received November 19, 2018; accepted February 7, 2019

Abstract: Knowledge Bases (KBs) enable engineers to capture knowledge in a formalized way. This formalization allows us to combine knowledge, thus creating the basis for smart factories while also supporting product and production system design. Building comprehensive and reusable KBs is still a challenge, though, especially for knowledge-intensive domains like engineering and production. To cope with the sheer amount of knowledge, engineers should reuse existing KBs. This paper presents a comprehensive overview of domain-specific KBs for production and engineering, as well as generic top-level ontologies. The application of such top-level ontologies offers new insights by integrating knowledge from various domains, stakeholders, and companies. To bridge the gap between top-level ontologies and existing domain KBs, we introduce an Intermediate Engineering Ontology (IEO).

Keywords: Smart factories, smart engineering, knowledge bases

Zusammenfassung: Wissensbasen (KBs) ermöglichen es Ingenieuren ihr Wissen zu formalisieren. Diese Formalisierung wiederum ermöglicht es Wissen zu bündeln und damit die Grundlage für intelligente Produktionssysteme zu schaffen. Gleichzeitig können die KBs auch die Entwicklung von Produkten und Produktionssystemen unterstützen. Der Schaffung umfassender und wiederverwendbarer KBs ist jedoch eine Herausforderung, insbesondere für wissensintensive Bereiche wie Entwicklung und Produktion. Um den enormen Wissensbestand dieser Bereiche nutzen zu können, sollten Ingenieure auf bestehende KBs zurückgreifen. Dieser Beitrag gibt einen umfassenden Überblick über domänenspezifische KBs für Entwicklung und Produktion sowie generische Top-Level-Ontologien. Die Anwendung solcher generischen Ontologien ermöglicht die Gewinnung neuer Erkenntnisse durch die Integration des Wissens verschiedener Domänen, Interessensgruppen und Unternehmen. Um die Lücke zwischen Top-Level-Ontologien und bestehenden domänenspezifischen KBs zu schließen, führen wir eine zwischengelagerte Ebene, die "Intermediate Engineering Ontology" (IEO), ein.

Schlagwörter: Intelligente Fabriken, Intelligentes Engineering, Wissensbasen

1 Introduction

In a world that continues to become more interlinked, we collect an ever increasing amount of information [1]. This impacts complex and knowledge-intensive professions especially, such as engineering and production. By leveraging the available information, unprecedented synergies can be achieved that help to build smart factories and support smarter engineering, too.

More specifically, Knowledge-Based Systems (KBSs) support engineers by making information about previous designs available, and by providing automatic feasibility feedback for new designs. Such feasibility feedback can help to integrate: diverse disciplines, e. g., mechanical and software engineering; different viewpoints along the product development process, e. g., product and process design; and companies and their suppliers. Additionally, the Knowledge Base (KB) can also serve as a basis for optimization, if several feasible designs exist. The accumulated knowledge can also be used later on in the development process. Parts of the developed KBs can be reused in agent based production systems, and KBSs provide the means to efficiently reconfigure entire plants.

However, the benefit from the information gathered is limited unless it is formalized and combined. Combining heterogeneous information from different domains, different stakeholders, and different companies still poses a major challenge. To address this challenge within the

***Corresponding author: Felix Ocker,** Technical University of Munich, Munich, Germany, e-mail: felix.ocker@tum.de
Christiaan J. J. Paredis, Clemson University, Clemson, USA, e-mail: paredis@clemson.edu
Birgit Vogel-Heuser, Technical University of Munich, Munich, Germany, e-mail: vogel-heuser@tum.de

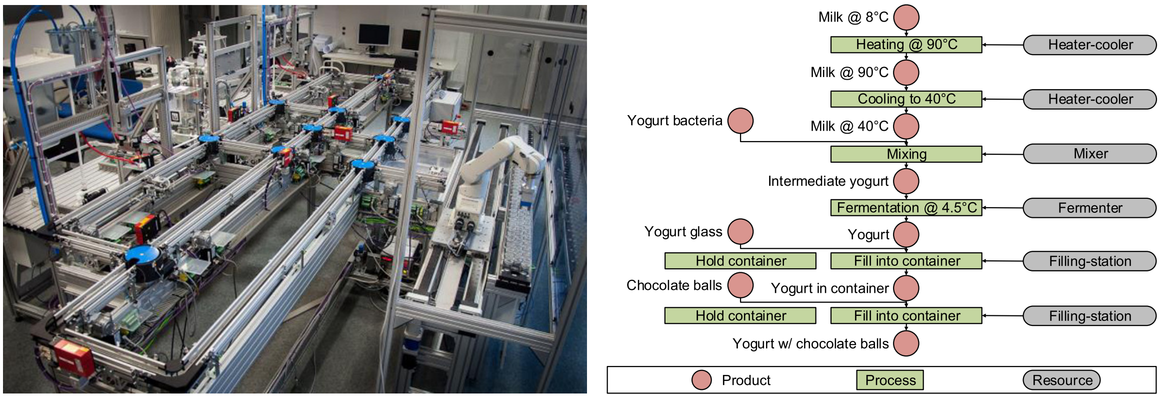


Figure 1: myJoghurt demonstrator (left) and functional process description of the yogurt production process (right).

engineering and production domain, we compare various established knowledge bases such as MAnufacturing Semantic's ONtology (MASON) [2] and Ontology for Computer Aided Process Engineering (OntoCAPE) [3] and analyze their compatibility. Also, we assess various generic, so-called top-level, ontologies regarding their suitability for the engineering and production domains. Based on this analysis, we derive an Intermediate Engineering Ontology (IEO) that allows us to combine domain-specific knowledge bases. Additionally, we align the IEO with the top-level Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) [4] to ensure that the knowledge base is well-formed and can be reused easily. With this paper, we contribute to the recent research that supports moving away from separate data silos to integrated knowledge bases.

2 Applications

The reusability of KBs can be increased by clarifying their purpose. This can be achieved by formulating competency questions [5, 6]. Competency questions are those questions that a KB should be able to answer.

The KB presented in this paper is designed for four applications, each refined by competency questions. These competency questions are made more accessible through the example of the myJoghurt demonstrator,[1] presented in Figure 1. The demonstrator consists of a logistics part and a procedural part. The logistics part comprises a storage unit, a robot, and a conveyor system, while the procedural part includes the equipment necessary to produce yogurt and two filling stations. Each of the filling stations can fill glasses with liquid and solid parts. The detailed process of the yogurt production and filling is described graphically via a functional process description in Figure 1.

First, we want to support engineers throughout the design process by providing feasibility feedback and automating the production planning. This is expressed by the two questions "*CQ1: Is this product feasible?*" and "*CQ2: In which order should the given resources execute the process steps?*". In the case of the myJoghurt demonstrator, the designer might want to check whether a certain chocolate ball fits through the dispenser. There might also be a restriction that the glass should always be filled with yogurt first. Additionally, interdisciplinary information should be checked for inconsistencies ("*CQ3: Which inconsistencies exist within the product's specification?*"). This can be as simple as checking whether the combined volume of the yogurt and, e. g., chocolate balls fits into the desired glass. Second, the KB considers supply chain planning, so that the capabilities of suppliers can be leveraged. Hence, with the KB, designers should be able to answer the questions "*CQ4: Which supplier can provide this subproduct?*" and "*CQ5: Which supplier has sufficient capacity to deliver in time?*". In the case of the yogurt production process, these two questions could be asked for chocolate balls, which are procured from a supplier. Third, the KB should also be usable for agent-based Cyber-Phyiscal Production Systems. Such autonomous but cooperative agents require individual KBs, which should be compatible. Only then can they answer questions like "*CQ6: Which resource is available to process this product?*". That way, the myJoghurt demonstrator can autonomously decide at which filling station the glass will be filled with yogurt. Also, in case of a local conveyor malfunction, the demonstrator could also find a new route through the logistics system. Fourth, the KB serves as information storage. This is expressed in the competency question "*CQ7: How did we*

1 http://i40d.ais.mw.tum.de/ last retrieved: February 12, 2019.

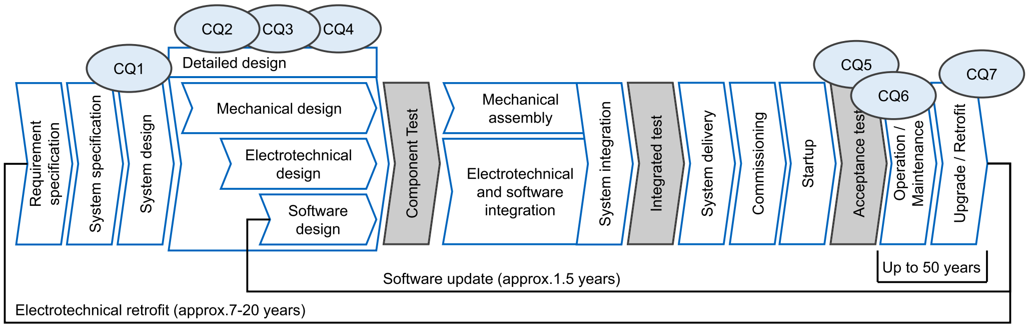


Figure 2: Competency questions throughout the product lifecycle, adapted from [7].

design previous plants?". That information can be used in two ways. On the one hand, the status of a plant can be represented, making it easier to maintain and possibly retrofit it. E. g., designers can more easily check in advance if a new sensor model can be used to replace an older one that is no longer available. Additionally, knowledge of previous designs can be reused when a similar plant is designed.

The competency questions show that the desired KB is relevant and usable throughout the entire lifecycle of consumer products as well as machines and plants. The questions' chronological order and their placement in the typical product lifecycle, adapted from [7], is depicted in Figure 2.

3 Established knowledge bases

Terminology should be clarified when discussing knowledge bases. Rowley [8] distinguishes data, information, and knowledge, which increase in value in this order. Value in this context is strongly linked to meaning, which in turn determines usability. Data are symbols, resulting from observation, which are useless without interpretation that turns them into information. In contrast, knowledge is defined as know-how, i. e., the "transformation of information into instructions" [8]. KBs can be classified according to their purpose. While reference ontologies accumulate knowledge of a specific domain, application ontologies have a more specific use case. Abstract top-level ontologies, finally, are the glue that holds everything together. Systems that make use of such a knowledge base and also support inference mechanisms to create new knowledge are known as KBSs [9].

This section gives an overview of domain-specific knowledge bases first. This is followed by an analysis of various top-level ontologies and their applicability to the production and engineering domains.

3.1 Domain-specific knowledge bases

Knowledge bases in engineering and production can be grouped into reference KBs and application KBs. While reference KBs serve as textbook-like collections of knowledge, application KBs are tailored to specific use cases.

Four reference KBs from the manufacturing domain are MAnufacturing Semantic's ONtology (MASON), Manufacturing Core Concepts Ontology (MCCO), Semantische Allianz für Industrie 4.0 (SemAnz40), and instant Foundry, Adaptive through Bits (iFAB). Combined, all of them form a basis for answering the competency questions defined. MASON [2] describes the Product Process Resource (PPR) structure, costs, and administrative entities. Even though MASON only presents a limited overview of manufacturing processes and other classes, it provides a well-defined structure. Manufacturing Core Concepts Ontology (MCCO) [10, 11, 12] consists of a set of universals and their key relationships relevant for product and process designers in manufacturing. Usman [11] emphasizes the combination of design and manufacturing features. Semantische Allianz für Industrie 4.0 (SemAnz40) helps to exchange product and process information in the context of smart factories [13]. This supports cooperation and collaboration of production systems. Lastly, the iFAB project resulted in a full-blown metamodel for manufacturing processes that includes an extensive process taxonomy [14, pp. 72 ff.] as well as cost and energy metrics for these processes [14, p. 71]. iFAB relies on a feature based approach to provide feedback if a product can be manufactured. For this, the product's final specification is required.

Apart from reference KBs, various application KBs have been developed for the manufacturing domain. Several application ontologies have been designed specifically to answer CQ2 and CQ6 for smart, agent based, factories. This way, flexibility of production systems is increased and their reconfiguration is enabled. Borgo et al.

[15, 16] created the ADAptive holonic COntrol aRchitecture for distributed manufacturing systems (ADACOR) ontology to model distributed manufacturing systems, including their modules and processes to support scheduling and monitoring. The ADAptive holonic COntrol aRchitecture for distributed manufacturing systems (ADACOR) ontology was aligned with the top-level ontology Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) to make sure that it is well-founded. Based on MASON, Alsafi and Vyatkin [17] present an agent based approach that automatically reconfigures manufacturing systems according to changes in requirements or the environment. The approach realizes high-level planning via the IEC 61499 standard. An agent based orchestration system is presented by Puttonen et al. [18]. They describe manufacturing web services in a domain ontology that by use of the classes product, equipment, and process. Using Sparql Protocol And RDF Query Language (SPARQL) queries, Puttonen et al. [18] check whether a product is finished without violating specific restrictions. Helbig et al. [19, 20] introduce Manufacturing System Dependency Model (MaSDeM), which focuses on modularity, flexibility, and reconfigurability. MaSDeM allows designers to describe products, manufacturing processes, electronics, software, and the dependencies in between them [19]. This description can be used to allocate distributed intelligence to subsystems, but also to support interdisciplinary collaboration during engineering. Similarly to MaSDeM, Ferrer et al. [21] map products, processes and resources in an ontology, to configure and analyze automation systems automatically. They present two Semantic Web Rule Language (SWRL) rules to identify necessary manufacturing processes for product variants. The model-based approach SkillPro [22] intends to automate process planning for manufacturing systems. SkillPro relies on the PPR structure to support automatic reconfiguration of production systems. Similarly, Harcuba and Vrba [23] present an ontology for flexible production systems, that results from the ARUM project. The ontology developed also uses the PPR structure and supports production scheduling.

There also exist application ontologies dedicated to supply chain management (CQ4 and CQ5), which also includes process sequencing (CQ2). The Manufacturing Service Description Language (MSDL) [24] supports agile manufacturing strategies for entire supply chains. Manufacturing Service Description Language (MSDL) describes services in detail, including process parameters, i. a. tolerances, weight, and size. Ameri et al. present an approach for discovering suitable suppliers [25] and classifying them based on rules [26]. Analogously to ADACOR, MSDL is aligned with the top-level ontology Basic Formal Ontology (BFO). This increases reusability of MSDL. Sarkar and Sormaz [27] map CAD product features to manufacturing processes via SWRL rules to derive manufacturing processes. They also present the reference ontology Semantically Integrated Manufacturing Planning Model (SIMPM) to formalize knowledge regarding manufacturing processes, but neglect resources. Legat et al. [28] optimize operation sequences using a formalized description of a plant's capabilities. Even though the focus is put on control software, the approach builds on a detailed plant model. Pre and post conditions for operations are modeled in the Object Constraint Language. HiTraP-AT [29] extends the original approach by optimizing field level automation software.

We also want to highlight four KBs from the process domain, namely Process Specification Language (PSL), Ontology for Computer Aided Process Engineering (OntoCAPE), Batch Process Ontology (BaPrOn) and Process Ontology (PrOnto). They can be consulted for CQ2, but they are also helpful regarding the other competency questions. Process Specification Language (PSL) is a robust and generic process specification developed by the National Institute of Standards and Technology (NIST) [30]. PSL Core provides axioms for activities, activity occurrences, timepoints and objects. OntoCAPE [3, 31, 32] is a well-founded, formalized and modular domain ontology from the process domain. During the development, Morbach et al. emphasized modularity and designed OntoCAPE with several layers to find a trade-off between usability and reusability. OntoCAPE finds an application in the Process Data Warehouse (PDW) [33], where it supports knowledge management in process design. The PDW makes design rationales reusable by tracing and recording decision-making procedures. The Batch Process Ontology (BaPrOn) is a reference ontology specifically for batch processes [34]. It adapts the ANSI/ISA-88 [35] standard for batch process control systems and has already been applied for scheduling-monitoring and decision making tasks [34]. Finally, Process Ontology (PrOnto) represents physical components of a process plant to support process planning of batch processes [36]. Lepuschitz et al. [37] benchmark selected ontologies from the batch processing domain concerning automation criteria, i. a. performance analysis, quality monitoring and process control on the controller level.

Table 1 gives an overview of selected domain KBs. The various foci of the KBs are mirrored in the choice of classes included and left out. E. g., from the KBs above, only iFAB considers tolerances, which shows its applicability for manufacturing. SemAnz40 on the other hand

Table 1: Comparison of selected domain ontologies.

IEO	ADACOR [15, 16]	iFAB [14]	MASON [2]	OntoCAPE/*PDW* [31, 33]	SemAnz40 [13]	SkillPro [38, 22]
product	product, raw material	product (input, output)	technological entity: raw material, assembly entity, etc.	raw material, intermediate, output product (by-, co-, core-, waste-product), *product object*	product	product
process	operation, milling etc.	(manufacturing) process	(manufacturing) operation	(batch) process, *process object/activity*	function/process	template-skill
resource	resource	resource	resource	plant item	technical resource	resource
capability	skill	capability			system behavior	resource skill
specification	process plan			*description object*		production skill
quality	quality	feature, constraint, metrics		property		attribute
quale	quale			value	variable	
machine	transporter, producer, mover	machine	machine resource	machine		machine resource
plant		facility		plant	structural hierarchy	
site			geographical resource		site	
unit	unit			unit		
employee		human resource			human resource	
operator	explicitly neglected	operator		user		
tool	tool	tool	tool			
fixture		fixture		fixture		

supports a more holistic view of clusters of companies, as it includes the universal *enterprise*, and OntoCAPE's focus on the chemical process industry is mirrored by the inclusion of universals for *piping*, *substance*, and *phase system*. The comparison of domain-specific KBs also shows that none of them can answer all competency questions defined. Hence, a combination of these KBs is required. In theory, this can be achieved by use of one of the top-level ontologies described in the following section.

3.2 Top-level ontologies

Ontologists have developed a variety of top-level ontologies to support the consistent development of domain ontologies. This section gives an overview of the most common top-level ontologies and evaluates them with regard to their applicability to the integration of product and process design. The evaluation is conducted at hand of six criteria, namely *expressivity*, *genericness*, *prevalence*, *size*, *availability* and *support*. An ontology's expressivity is mostly influenced by its focus, i. e., it is fit to describe the PPR structure and other relevant notions. Genericness on the other hand means that there are little limitations to how the ontology can be extended. Prevalence describes the ontology's use in practice and is a good measure for its maturity. An upper ontology's size is relevant, as it greatly influences reasoning and querying performance. Concerning performance, a large upper ontology is only acceptable if its expressivity is also high. Availability including licensing is crucial for use. Support finally means that the ontology is well documented. The first two criteria expressivity and genericness should be intrinsic to any top-level ontology. However, philosophical controversies may lead to fundamental design decisions which in turn can cause limitations.

Sowa's ontology [39] is not included in this list since no documented applications have been developed [40]. It deserves being mentioned, though, as it is a lean top-level ontology that inspired many existing upper ontologies [40].

The Basic Formal Ontology (BFO) [41, 42] is a well founded top-level ontology that focuses on universals in reality [42, p. 39]. This has the drawback that e. g., planned products, processes or resources can only be included in an ontology as plans. Having to distinguish between planned and already existing entities makes some aspects in product and process development unnecessarily complex and leads to an overhead. Exemplary, it is irrelevant for a feasibility check whether a certain resource already exists if the designer can assume that it will be available at the start of production. Also, BFO deliberately does not support mathematical notions, and the representation of units is still under development. However, BFO provides roles, which may be a useful feature. Axiomatization in the base version is realized via *subClassOf* relations and the definition of disjoint classes only. BFO ensures genericness by not containing any "representations of physical, chemical, biological, psychological, or other types of entities which would properly fall within the domains of the special sciences" [43]. This genericness enables over 130 public-domain ontologies to build on BFO [42, p. 39]. Even though these are primarily from the biological and biomedical domain, e. g., the MSDL [44] was also aligned with BFO. The associated domain ontologies show BFO's prevalence. BFO's base version includes only 35 classes, which is smaller than both DOLCE and Standard Upper Merged Ontology (SUMO). Hence, BFO is "more manageable as an artifact designed for purposes of ontological engineering" [43]. To reach the same expressivity as e. g., DOLCE, extensions such as the relations ontology (RO) have to be loaded, though. BFO is published under a creative commons license and is available online.[2] Supported specification languages are Web Ontology Language (OWL), OBO and CLIF. BFO is still under development, with new versions being released whenever sensible. To help ontologists that use BFO, tools for automatic upgrades are provided. Support also includes extensions to BFO such as the relations ontology[3] (RO) or the information artifact ontology (IAO) [45].

Cyc is a "large knowledge base containing a store of formalized background knowledge suitable for supporting reasoning in a variety of domains" [46]. Mascardi et al. [40] emphasize the focus on "facts, rules of thumb and heuristics for reasoning about objects and events of everyday life". Cyc is structured into three parts, namely an upper, a middle and a lower ontology [46]. While the upper ontology includes generic terms i. a. *event* or *simulation*, the middle ontology captures terms that are widely used, but not necessarily applicable to all domains, i. a. *SocialGathering*. The lower Cyc ontology finally includes domain specific terms, i. a. *ChemicalReaction*. Cyc's genericness decreases from the upper to the lower level, where only the upper level is on the same level of abstraction as e. g., BFO.

2 https://github.com/BFO-ontology/BFO last retrieved: February 12, 2019.

3 http://obofoundry.org/ontology/ro.html last retrieved: February 12, 2019.

Applications of Cyc are diverse and include pharmaceutical thesaurus management, semiconductor yield management and clinical trial and reporting support.[4] The full version of Cyc describes more than 250 thousand terms including almost 15 thousand predicates [46]. Due to its size, Cyc's formal consistency is hard to ensure [46], which is a major limitation. Apart from a commercial license, Cycorp also provides a free research license. The open source version OpenCyc is no longer available as of 2017.Cyc uses its own language CycL and support is available for the commercial and the research version.

The Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) [4, 47, 48] is part of the WonderWeb foundational ontologies library and "aims at capturing the ontological categories underlying natural language and human commonsense" [4]. In contrast to BFO, it relies on possible worlds [42, p. 39] and is an ontology of particulars [4]. This means, it does not have the same limitations as BFO concerning overheads in product and process design. Compared to BFO, DOLCE offers an intuitive way of integrating units and possesses an extensive axiomatization. Furthermore, DOLCE supports functional modeling [49]. Being a proper top-level ontology, DOLCE has about the same level of genericness as BFO. Applications of DOLCE are found mostly in biology and social science [42, p. 39], but Borgo and Leitão [15] also used DOLCE for a manufacturing ontology based on the ADACOR architecture. There exist three different versions of DOLCE, with DOLCE Lite being the leanest one, as it contains 37 classes and 70 object properties. There is also an extended version for descriptions and situations (DNS) as well as DOLCE Lite Plus (DLP), which contains 208 classes and 313 object properties. DLP's expressivity exceeds the one of BFO, though, as it includes entities from BFO's IAO, too. Since DOLCE is modular, ontologists can decide to only import relevant parts, which compensates for DOLCE being bigger than BFO. DOLCE is available for free[5] in OWL, with the latest build (397) existing unchanged since 2006.

Similar to Cyc, the General Formal Ontology (GFO) [50, 51] provides a three-layered meta-ontological architecture consisting of an abstract top level, an abstract core level and a basic level. General Formal Ontology (GFO) includes classes for objects, processes, time and space, roles, functions, facts and situations as well as properties [40]. Even though GFO was originally designed for the medical, biological and biomedical domain, it is generic enough to also fit the needs of economists and sociologists [51]. Just like with DOLCE, there exist different versions of GFO, too. The basic version comprises 45 classes and 41 object properties while the full version includes 78 classes and 67 object properties. Both are available for free[6] as OWL files. The latest full version was published in 2006 with the latest basic version following in 2008.

gist [52] is a minimal upper ontology focusing on business ontology notions. It includes 19 modules and relies on twelve fundamental classes. In contrast to e. g., BFO and DOLCE, these underlying classes are quite specific and intuitively understandable terms i. a. *Intention* or *Organization*. gist is axiomatized beyond subclass relations and disjoint, thus supporting more advanced reasoning. Applications of gist are diverse and range from R&D to investment banking to materials management. In total, gist is still of medium size, including 130 classes, 99 object properties and 21 datatype properties. The current version 7.5 from 2017 is available online[7] in OWL under a creative commons license.

Based on the ISO 15926 standard, an upper ontology was developed, too [53, 54, 55]. Its primary goal is to facilitate the exchange and reuse of complex plant and project information [55, p. 1] with a focus on the process industry [54]. The ISO 15926 ontology supports four-dimensionalism, so that space and time can be depicted appropriately. The ontology is not a pure upper ontology, but also includes domain specific knowledge from the production domain. Examples are classes relevant for design, engineering, procurement, building, commissioning, operation, maintenance and decommissioning. According to Morbach et al. [31], the standard's primary applications are in the area of data management. The ISO 15926 ontology was first applied to plant operation in the oil and gas industry [55]. Its focus is still on the process domain, even though it also includes a generic upper ontology. I.e. it is not a top-level ontology in the same sense as BFO or DOLCE. The version available online[8] includes 203 classes and 106 object properties. Considering the ontology's expressivity in the process domain, the size is still moderate. This OWL file from 2014 is freely available online, but the documentation within the standard is commercial. The use of the ISO 15926 ontology as an upper ontolgy is heavily criticized in [56]. According to Smith [56], the ISO 15926 ontology has major defects concerning intelligibility, open-

4 http://www.opencyc.org/ last retrieved: February 12, 2019.

5 http://www.loa.istc.cnr.it/old/DOLCE.html last retrieved: February 12, 2019.

6 http://www.onto-med.de/ontologies/index.jsp last retrieved: February 12, 2019.

7 https://semanticarts.com/gist/ last retrieved: February 12, 2019.

8 http://15926.org/standards/ last retrieved: February 12, 2019.

ness, reuse, coherence and compositional term construction [56].

PROTo ONtology (PROTON) [57] was designed with regard to four principles. While maintaining domain independence, PROTON includes light-weight logical definitions and is aligned with popular standards. Finally, it shall provide a good coverage of named entities and concrete domains [57, p. 1]. To realize these objectives, PROTON consists of four modules, namely System, Upper, Top and KM. The top module resembles gist and contains generic classes i. a. entity, event and person, while the extension also includes very specific terms like *OilField*. Overall, PROTON's top level is not as generic as BFO or DOLCE, though. PROTON is used in a variety of research projects,[9] with many of them being related to journalism and society. The top module[10] consists of only 25 classes and 77 properties, while its extension provides 488 classes and 115 properties. Descriptions are available for most entities and both ontologies are available under a creative commons license.

The Standard Upper Merged Ontology (SUMO) [58, 59] was developed by the IEEE's standard upper ontology working group and "provides definitions for general-purpose terms and acts as a foundation for more specific domain ontologies" [58]. Mascardi et al. [40] describe SUMO as a combination of the top level, a mid-level-ontology (MILO) and several domain ontologies. The base ontology not only includes mereotopology and temporal notions, but also supports class theory and numerics. SUMO's top level distinguishes physical and abstract entities, with the former including everything that has a position in spacetime [58]. On a lower level, SUMO contains e. g., biological terms, which means it is "not a top-level ontology in the same sense as BFO and DOLCE" [42, p. 39]. SUMO has been applied to various domains and over 100 appliciation papers have been published [40]. Also, SUMO is the largest formal public ontology today with approximately 25 thousand terms and 80 thousand axioms, when all domain ontologies are combined. Available domain ontologies range from food to law to weather. There also exists a domain ontology for engineering[11] which misses classes crucial for production, e. g., *Product* or *Resource*, though. SUMO is available online in OWL and SUO-KIF under a GNU public license.[12]

The Unified Foundational Ontology (UFO) [60, 61] is a "philosophically and cognitively well-founded formal ontology" [61]. It was designed specifically to serve as a foundational theory for conceptual modeling [61]. Unified Foundational Ontology (UFO) is based on GFO and makes a fundamental distinction between particulars and universals. This is a contrast to DOLCE, which is an ontology of particulars. UFO and DOLCE share their view of qualities and quality dimensions, though. UFO consists of an ontology for endurants, one for perdurants, one for social universals and one for services. The first three include 70 classes. So far, only two public domain ontologies are based on UFO, including a transport network ontology. For UFO only a specification including some descriptions is available online.[13] A full description of the included classes and properties is given in [60].

Yet Another More Advanced Top-level Ontology (YAMATO) [62] is heavily influenced by the top-level ontologies DOLCE, BFO, GFO, SUMO and Cyc. It combines most of their features and explicitly addresses qualities and quantities, representation and the distinction between processes and events, where some of the existing top-level ontologies fall short [62]. However, criticism towards YAMATO includes that it is too large and complex [62]. In comparison to BFO, some ideas fit the needs of developers from the engineering domain better. This includes the way qualities and units are handled. Concerning reasoning, YAMATO is well founded, as a first order formalization is available, cp. e. g., [63]. YAMATO describes 925 universals ranging from a generic *particular* to the quite specific *hill* or *rational-mind* and 183 object properties, also including very specific ones i. a. *has-hand*. These examples show that YAMATO comprises a variety of classes which are inadequately specific for a top-level ontology and are irrelevant for some domains. Applications of YAMATO are diverse, including medicine, learning and instructional theories as well as genomics. YAMATO is available online[14] as an OWL file.

Table 2 presents a qualitative overview of the evaluation of the described top-level ontologies in alphabetical order.

BFO, DOLCE and YAMATO achieve the best scores when evaluated concerning the criteria relevant for the approach presented. However, as Herre [51] states, "one may doubt whether a final and uniquely determined top level ontology can ever be achieved". One way to cope with this

9 http://ontotext.com/proton/ last retrieved: February 12, 2019.
10 http://www.ontotext.com/proton/protontop.html last retrieved: February 12, 2019.
11 https://github.com/ontologyportal/sumo/blob/master/engineering.kif last retrieved: February 12, 2019.
12 http://www.adampease.org/OP/ last retrieved: February 12, 2019.
13 http://ontology.com.br/ last retrieved: February 12, 2019.
14 http://download.hozo.jp/onto_library/upperOnto.htm last retrieved: February 12, 2019.

Table 2: Qualitative evaluation of top-level ontologies' applicability for the IEO.

	Expressivity	Genericness	Prevalence	Size	Availability	Support
BFO	–	+	+	+	+	+
Cyc	+	o	o	–	–	o
DOLCE	+	+	+	+	+	+
GFO	o	o	–	+	+	+
gist	–	–	+	+	+	o
ISO 15926	+	–	–	+	–	–
PROTON	o	o	o	+	+	o
SUMO	o	+	+	–	+	o
UFO	+	+	–	+	–	o
YAMATO	+	o	+	–	+	+

limitation and also with the variety of existing top-level ontologies is to use mappings. This allows us to indirectly combine domain ontologies that were aligned with different upper ontologies. Temal et al. [64] present a mapping between BFO and DOLCE. Similarly, Seppälä [65] map the lexical database WordNet[15] [66] via DOLCE indirectly to BFO. Another partial mapping exists for DOLCE and GFO [50, pp. 57 ff.].

Based on the evaluation of the upper ontologies, as well as the possibilities of mapping and thus exchange with other upper ontologies, we chose DOLCE as an appropriate top-level ontology for this work.

4 Bridging the gap between domain-specific knowledge bases and top-level ontologies

In the following, we introduce the Intermediate Engineering Ontology (IEO). The IEO serves as a connector between existing domains-specific KBs and the top-level ontology DOLCE, thus creating a trade-off between usability and reusability. Also, the IEO integrates product and process design, with the competency questions that were defined earlier in mind. By formalizing this knowledge, automatic analyses are enabled via reasoners and queries. An overview of the IEO's universals and properties is presented in Figure 3.

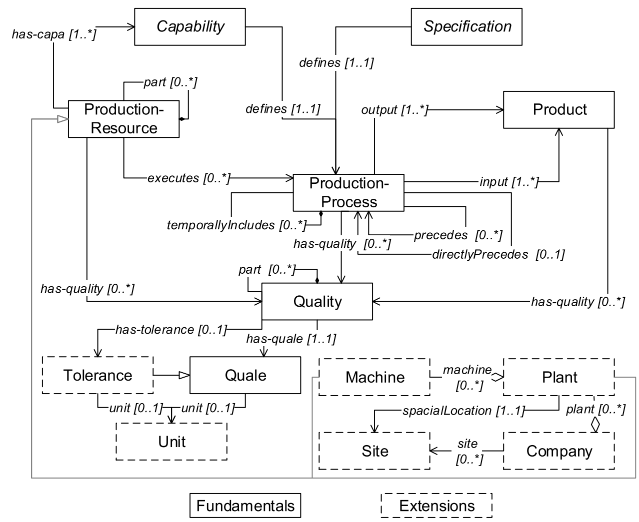


Figure 3: Overview of the IEO.

The IEO revolves around the Product Process Resource (PPR) structure, which can be refined by the use of *qualities* and *quales*. This reification is in line with DOLCE. Everything listed in a bill of materials (BOM), even bulk materials such as adhesives, can be represented by the universal *product*. Analogously, all elements of the bill of processes (BOP) are represented as a *production-process* and the entire bill of resources (BOR) can be described by use of the universal *production-resource*. The two abstract universals *capability* and *specification* are intended to model the capabilities of resources and the specification of the designer, respectively. Both of them define a *production-process*, which is connected to its *input* and *output* products. While *capabilities* describe the *processes* and related *products* a *resource* can handle, a *specification* defines the product designers' restrictions towards *processes* and *products*. Both *capabilities* and *specifications* define exactly one *production-process*, but a *resource* may possess several *capabilities*.

These fundamental universals are extended by the universals shown with dashed lines. To increase expressiveness, we distinguish between *plants* and *machines*, both of which are *production-resources*. In an engineering context, *units* and *tolerances* are also required. Finally, we also include a universal *company*, which corresponds to the class *enterprise* used in SemAnz40. A *company* can be described by associating its *plants* with *sites*, i. e., physical locations.

15 https://wordnet.princeton.edu/ last retrieved: February 12, 2019.

Further extensions of the fundamental universals include *human workers* and more detailed parts of *production-resources*, i. a. *tools*, *fixtures*, and *auxiliary resources*. Regarding *human workers*, *designers* and *operators* should be distinguished, each with an individual set of *skills* and different responsibilities.

To maintain the desired genericness of the universals, they are only formalized as depicted in Figure 3.

4.1 Aligning the IEO with DOLCE

Table 3 displays how we align the universals of the IEO to DOLCE. Hereby, details regarding DOLCE [16, 4, 47] are of great help, as well as the description of how ADACOR is aligned with DOLCE [15].

Table 3: Aligning relevant universals with DOLCE.

IEO universal	DOLCE universal
product	derived from edns:non-agentive-physical-object
production-process	derived from dol:process
production-resource	derived from edns:non-agentive-physical-object
capability	derived from edns:non-agentive-social-object
specification	derived from edns:non-agentive-social-object
quality	exists as dol:quality, which is refined by dol:physical-quality and dol:abstract-quality
quale	exists as dol:quale
machine	derived from :production-resource
company	derived from soc:organization
plant	derived from :production-resource
site	derived from dol:space-region
tolerance	derived from dol:quale
unit	exists as common:measurement-unit

Trying to find usability-reusability trade-off, we choose to formalize all universals only to the degree depicted in Figure 3. However, all universals can easily be refined, e. g., specific processes such as manufacturing or batch processes can be organized in categories derived from the class *production-process*.

Similarly, all object properties are aligned with DOLCE, cp. Table 4.

Table 4: Aligning relevant object properties with DOLCE.

IEO object property	DOLCE object property
defines	derived from dol:immediate-relation
directly-precedes	derived from tem:precedes
has-capa	derived from dol:immediate-relation
has-quale	exists as dol:has-quale
has-quality	exists as dol:has-quality
has-tolerance	derived from dol:r-location
input	derived from dol:participant
output	derived from dol:participant
owns	derived from dol:part
part	exists as dol:part
precedes	exists as tem:precedes
temporally-includes	exists as tem:temporally-includes
unit	exists as common:unit

In addition to these object properties, we introduced the three functional data properties *has-value*, *has-upper-bound* and *has-lower-bound* to assign numeric values to quales. All three are defined as elementary, because DOLCE does not include appropriate datatype properties.

As shown, the IEO presented is aligned to the top-level ontology DOLCE. This confirms that DOLCE is an appropriate choice for the production context.

4.2 Comparing the IEO with domain specific KBs

In this section, we compare the IEO with the domain-specific KBs presented. Due to a lack of information, we neglected MaSDeM [20, 19], Process Ontology (PrOnto) [36], and SIMPM [27].

The foci of these domain-specific KBs is mirrored in the more specific classes intentionally neglected by the IEO, cp. also the last row of Table 1. Even though not all KBs adhere to the reification proposed by DOLCE, but use properties instead, they can still be integrated using rules.

Table 5 gives an overview of the comparison of the IEO's universals and the classes used in the KBs presented earlier.

Table 5 reveals that all KBs rely on the PPR structure, making it easier to integrate them. Table 5 also shows how existing domain KBs complement each other and how they can be integrated with the IEO presented. This illustrates how the IEO manages to build on existing knowledge,

Table 5: Comparison of the IEO with existing domain ontologies.

IEO	ADACOR [15, 16]	ARUM [23]	BaPrOn [34]	HiTraP [29]	iFAB [14]	MASON [2]	MCCO/MRO [10, 12]	MSDL [24]	OntoCAPE [31, 33]	SemAnz40 [13]	SkillPro [22, 38]
product	X	X	X	X	X	X	X	X	X	X	X
production-process	X	X	X	X	X	X	X	X	X	X	X
production-resource	X	X	X	X	X	X	X	X	X	X	X
capability	X	X			X		X	X		X	X
specification	X	X	X				X		X		X
quality	X				X		X		X		X
quale	X								X	X	
machine	X		X		X	X		X	X		X
company			X				X			X	
plant				X	X		X		X	X	
site			X			X	X			X	
tolerance					X						
unit	X								X		

while staying potentially reusable across applications at the same time.

5 Summary and outlook

This paper gives a comprehensive overview of existing domain-specific KBs and abstract top-level ontologies. Since there is a gap between the usability-oriented application ontologies and the reuse-oriented top-level ontologies, we present an Intermediate Engineering Ontology (IEO) that helps to combine the two. The IEO is designed to answer several competency questions throughout the lifecycle of a product, be it a consumer product or an entire plant.

Starting with this paper as an overview, we intend to refine and assess the competency questions identified in greater detail to evaluate the opportunities and boundaries of the IEO. This is supported by integrating standardization approaches for domain ontologies. One example of the production domain is presented by Hildebrandt et al. [5]. A combination with less formal databases such as eCl@ss or the IEC 61360 Common Data Dictionary [67] should also be pursued. Even though process chains can already be represented formally, behavior descriptions should be discussed in greater detail. Additionally, the integration of online information into the ontology should be investigated. Such information can range from the production status of products to sensor information of the plant. A major challenge that arises from combining such extensive KBs is scalability. The scalability issue is intensified when the KBs are extended by the amount of information available in industrial settings. This calls for two technological developments. First, we will need more efficient reasoning methods to cope with the sheer amount of information. Secondly, designers require frameworks tailored specifically to the reuse of existing methodologies, so that established tools can still be used.

Because what we think of as knowledge changes continuously, the process of ontology design is open-ended [42]. However, we believe that the combination of top-level ontologies and domain-specific reference ontologies with intermediate application ontologies provides a good tradeoff regarding usability and reusability. In summary, we expect the IEO to perform well in the process of combining knowledge to gain new insights.

Funding: The authors thank the German Research Foundation (DFG) for funding the project CRC 768.

References

1. B. Marr, "How Much Data Do We Create Every Day? The Mind-Blowing Stats Everyone Should Read," 2018. [Online]. Available: https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read/.
2. S. Lemaignan, A. Siadat, J.-Y. Dantan and A. Semenenko, "MASON: A Proposal For An Ontology Of Manufacturing Domain," in *Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications*. Prague, Czech Republic: IEEE, 2006, pp. 195–200.
3. J. Morbach, M. Theißen and W. Marquardt, "Integrated Application Domain Models for Chemical Engineering," in *Collaborative and Distributed Chemical Engineering. From Understanding to Substantial Design Process Support*, M. Nagl and W. Marquardt, Eds. Berlin, Heidelberg: Springer, 2008, pp. 169–182.
4. A. Gangemi, N. Guarino, C. Masolo, A. Oltramari and L. Schneider, "Sweetening Ontologies with DOLCE," in *Knowledge Engineering and Knowledge Management, Sigüenza, Spain*. Berlin, Heidelberg: Springer, 2002, pp. 166–181.
5. C. Hildebrandt, S. Törsleff, B. Caesar and A. Fay, "Ontology Building for Cyber-Physical Systems: A domain expertcentric approach," in *CASE*. Munich, Germany: IEEE, 2018.

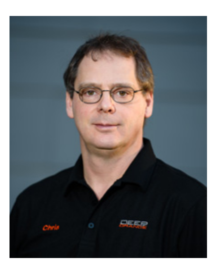
6. N. F. Noy, D. L. McGuinness *et al.*, "Ontology development 101: A guide to creating your first ontology," Stanford knowledge systems laboratory technical report KSL-01-05 and Stanford medical informatics technical report SMI-2001-0880, Stanford, USA, Tech. Rep., 2001.
7. B. Vogel-Heuser and S. Rösch, "Applicability of technical debt as a concept to understand obstacles for evolution of automated production systems," in *SMC*. IEEE, 2015, pp. 127–132.
8. J. Rowley, "The wisdom hierarchy: representations of the dikw hierarchy," *Journal of information science*, vol. 33, no. 2, pp. 163–180, 2007.
9. R. G. Smith, "Knowledge-based systems: Concepts, techniques, examples," *Canadian High Technology Show*, 1985.
10. Z. Usman, R. I. M. Young, N. Chungoora, C. Palmer, K. Case and J. Harding, "A Manufacturing Core Concepts Ontology for Product Lifecycle Interoperability," in *IWEI*, M. van Sinderen and P. Johnson, Eds. Berlin, Heidelberg: Springer, 2011, pp. 5–18.
11. Z. Usman, "A manufacturing core concepts ontology to support knowledge sharing," Ph.D. dissertation, Loughborough University, 2012.
12. Z. Usman, R. Young, N. Chungoora, C. Palmer, K. Case and J. Harding, "Towards a formal manufacturing reference ontology," *International Journal of Production Research*, vol. 51, no. 22, pp. 6553–6572, 2013.
13. C. Hildebrandt, A. Scholz, A. Fay, A. F. De, T. Schröder, T. Hadlich, C. Diedrich, M. Dubovy, C. Eck and R. Wiegand, "Semantic Modeling for Collaboration and Cooperation of Systems in the production domain," in *ETFA*. Limassol, Cyprus: IEEE, 2017.
14. S. N. Melkote, "Development of iFAB Manufacturing Process and Machine Library," Georgia Institute of Technology, Tech. Rep., 2012.
15. S. Borgo and P. Leitão, "The Role of Foundational Ontologies in Manufacturing Domain Applications," in *OTM*, R. Meersman and Z. Tari, Eds. Berlin, Heidelberg: Springer, 2004, pp. 670–688.
16. S. Borgo and P. Leitão, "Foundations for a Core Ontology of Manufacturing," in *Ontologies*. Boston, USA: Springer US, 2007, pp. 751–775.
17. Y. Alsafi and V. Vyatkin, "Ontology-based reconfiguration agent for intelligent mechatronic systems in flexible manufacturing," *Robotics and Computer-Integrated Manufacturing*, vol. 26, no. 4, pp. 381–391, 2010.
18. J. Puttonen, A. Lobov and M. Lastra, "Semantics-Based Composition of Factory Automation Processes Encapsulated by Web Services," *TII*, vol. 9, no. 4, pp. 2349–2359, 2012.
19. T. Helbig, E. Westkamper and J. Hoos, "Identifying automation components in modular manufacturing systems: A method for modeling dependencies of manufacturing systems," in *ETFA*. IEEE, 2014, pp. 1–8.
20. T. Helbig, S. Erler, E. Westkämper and J. Hoos, "Modelling Dependencies to Improve the Cross-domain Collaboration in the Engineering Process of Special Purpose Machinery," *Procedia CIRP*, vol. 41, pp. 393–398, 2016.
21. B. R. Ferrer, B. Ahmad, A. Lobov, D. A. Vera, J. L. M. Lastra and R. Harrison, "An approach for knowledge-driven product, process and resource mappings for assembly automation," in *CASE*. IEEE, 2015, pp. 1104–1109.
22. I. M. Criado, K. Aleksandrov, S. E. Navarro, K. Georgoulias, R. Henßen, J. Pfrommer, F. Ubis and D. Štogl, "SkillPro Deliverable D2.1.0," TECNALIA, Tech. Rep., 2014.
23. O. Harcuba and P. Vrba, "Ontologies for flexible production systems," in *ETFA*. Luxembourg, Luxembourg: IEEE, 2015, pp. 1–8.
24. F. Ameri and D. Dutta, "An Upper Ontology for Manufacturing Service Description," in *IDETC/CIE*. ASME, 2006, pp. 651–661.
25. F. Ameri, C. McArthur, B. Asiabanpour and M. Hayasi, "A Web-based Framework for Semantic Supplier Discovery for Discrete Part Manufacturing," in *Transactions of the NAMRI/SME*, 2011.
26. F. Ameri and C. McArthur, "Semantic rule modelling for intelligent supplier discovery," *Int J Comput Integ M*, vol. 27, no. 6, pp. 570–590, 2014.
27. A. Sarkar and D. Sormaz, "Foundation Ontology for Distributed Manufacturing Process Planning," in *CIE*. Charlotte, USA: ASME, 2016.
28. C. Legat, D. Schütz and B. Vogel-Heuser, "Automatic generation of field control strategies for supporting (re-)engineering of manufacturing systems," *J Intell M*, vol. 25, no. 5, pp. 1101–1111, 2014.
29. C. Legat and B. Vogel-Heuser, "A configurable partial-order planning approach for field level operation strategies of PLC-based industry 4.0 automated manufacturing systems," *Engineering Applications of Artificial Intelligence*, vol. 66, pp. 128–144, 2017.
30. C. I. Schlenoff, R. W. Ivester and A. Knutilla, "A Robust Ontology for Manufacturing Systems Integration," in *International Conference on Engineering Design and Automation, Maui, Hawaii*, 1998.
31. J. Morbach, A. Wiesner and W. Marquardt, "OntoCAPE – A (re)usable ontology for computer-aided process engineering," *Comput Chem Eng*, vol. 33, no. 10, pp. 1546–1556, 2009.
32. J. Morbach, "A Reusable Ontology for Computer-Aided Process Engineering," Ph.D. dissertation, Rheinisch-Westfälische Technische Hochschule Aachen, 2009.
33. S. C. Brandt, J. Morbach, M. Miatidis, M. Theißen, M. Jarke and W. Marquardt, "An ontology-based approach to knowledge management in design processes," *Comput Chem Eng*, vol. 32, no. 1-2, pp. 320–342, 2008.
34. E. Muñoz, A. Espuña and L. Puigjaner, "Towards an ontological infrastructure for chemical batch process management," *Comput Chem Eng*, vol. 34, no. 5, pp. 668–682, 2010.
35. International Society of Automation, "ANSI/ISA-88," Tech. Rep., 2010.
36. W. Lepuschitz, B. Groessing, E. Axinia and M. Merdan, "Phase Agents and Dynamic Routing for Batch Process Automation," in *HoloMAS, Prague, Czech Republic*. Springer, Berlin, Heidelberg, 2013, pp. 37–48.
37. W. Lepuschitz, A. Lobato-Jimenez, E. Axinia and M. Merdan, "A Survey on Standards and Ontologies for Process Automation," in *HoloMAS, Valencia, Spain*. Springer, Cham, 2015, pp. 22–32.
38. M. Schleipen, "AutomationML to describe skills of production plants based on the PPR concept," in *AutomationML user conference*, Blomberg, 2014.
39. J. F. Sowa, *Knowledge representation: logical, philosophical, and computational foundations*. Pacific Grove, USA: Brooks Cole Publishing Co., 1999.

40. V. Mascardi, V. Cordì and P. Rosso, "A Comparison of Upper Ontologies," in *WOA*, 2007, pp. 55–64.
41. B. Smith, "Basic Formal Ontology 2.0," Tech. Rep., 2015.
42. R. Arp, B. Smith and A. D. Spear, *Building ontologies with basic formal ontology*. MIT Press, 2015.
43. B. Smith, "On Classifying Material Entities in Basic Formal Ontology," in *Interdisciplinary Ontology Meeting*. Keio University Press, 2012, pp. 1–13.
44. F. Ameri, "Manufacturing Supply Chain Ontology - Experiences with BFO," in *IDETC/CIE*. Cleveland, USA: ASME, 2017.
45. W. Ceusters and B. Smith, "Aboutness: Towards Foundations for the Information Artifact Ontology," 2015. [Online]. Available: https://philpapers.org/rec/CEUATF.
46. C. Matuszek, J. Cabral, M. Witbrock and J. Deoliveira, "An introduction to the syntax and content of Cyc," in *Spring Symposium on Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering*. AAAI, 2006, pp. 44–49.
47. C. Masolo, S. Borgo, A. Gangemi, N. Guarino, A. Oltramari and L. Schneider, "The WonderWeb Library of Foundational Ontologies," ISTC-CNR, Tech. Rep., 2003.
48. S. Borgo and C. Masolo, "Foundational Choices in DOLCE," in *Handbook on Ontologies*. Berlin, Heidelberg: Springer, 2009, pp. 361–381.
49. S. Borgo, M. Carrara, P. Garbacz and P. E. Vermaas, "A Formalization of Functions as Operations on Flows," *JCISE*, vol. 11, no. 3, 2011.
50. H. Herre, B. Heller, P. Burek, R. Hoehndorf, F. Loebe and H. Michalek, "General Formal Ontology (GFO) Part I: Basic Principles," IMISE, Leipzig, Tech. Rep., 2006.
51. H. Herre, "General Formal Ontology (GFO): A Foundational Ontology for Conceptual Modelling," in *Theory and Applications of Ontology: Computer Applications*. Dordrecht: Springer Netherlands, 2010, pp. 297–345.
52. D. McComb, "A Minimalist Upper Ontology – Semantic Arts," 2010. [Online]. Available: https://semanticarts.com/articles/semantics-and-ontologies/a-minimalist-upper-ontology/.
53. R. Batres, M. West, D. Leal, D. Price, K. Masaki, Y. Shimade, T. Fuchino and Y. Naka, "An upper ontology based on ISO 15926," *Comput Chem Eng*, vol. 31, no. 5-6, pp. 519–534, 2007.
54. J. W. Klüwer, M. G. Skjaeveland and M. Valen-Sendstad, "ISO 15926 templates and the Semantic Web," in *W3C Workshop on Semantic Web in Energy Industries; Part I: Oil & Gas*. W3C, 2008.
55. Fiatech, *An Introduction to ISO 15926*, 2011.
56. B. Smith, "Against idiosyncrasy in ontology development," *Frontiers in Artificial Intelligence and Applications*, vol. 150, 2006.
57. I. Terziev, A. Kiryakov and D. Manov, "D1.8.1 Base upper-level ontology (BULO) Guidance," Ontotext Lab, Sirma Group, Tech. Rep., 2005.
58. I. Niles and A. Pease, "Towards a Standard Upper Ontology," in *Formal Ontology in Information Systems*. Ogunquit, USA: ACM, 2001.
59. I. Niles and A. Pease, "Origins of The IEEE Standard Upper Ontology," *Working Notes of the IJCAI-2001 Workshop on the IEEE Standard Upper Ontology*, vol. 17, pp. 4–10, 2001.
60. G. Guizzardi, "Ontological foundations for structural conceptual models," Ph.D. dissertation, University of Twente, 2005.
61. G. Guizzardi and G. Wagner, "Using the Unified Foundational Ontology (UFO) as a Foundation for General Conceptual Modeling Languages," in *Theory and Applications of Ontology: Computer Applications*. Dordrecht, Netherlands: Springer Netherlands, 2010, pp. 175–196.
62. R. Mizoguchi, "YAMATO: Yet Another More Advanced Top-level Ontology," in *Advances in Ontologies*, K. Taylor, T. Meyer and M. Orgun, Eds. Adelaide, Australia: IAOA, 2010, pp. 1–15.
63. S. Borgo, "An ontological approach for reliable data integration in the industrial domain," *Comput Ind*, vol. 65, no. 9, pp. 1242–1252, 2014.
64. L. Temal, A. Rosier, O. Dameron and A. Burgun, "Mapping BFO and DOLCE," *Studies in Health Technology and Informatics*, vol. 160, pp. 1065–1069, 2010.
65. S. Seppälä, "Mapping WordNet to Basic Formal Ontology using the KYOTO ontology," in *ICBO, Lisbon, Portugal*, 2015.
66. C. Fellbaum, *WordNet: an electronic lexical database*. MIT Press, 1998.
67. "IEC 61360-4 – Common Data Dictionary," Tech. Rep., 2015.

Bionotes

Felix Ocker
Technical University of Munich, Munich, Germany
felix.ocker@tum.de

Felix Ocker, M.Sc., is a graduate research assistant and Ph.D. student with the Institute of Automation and Information Systems at the Technical University of Munich. His research focuses on knowledge formalization and inconsistency management in interdisciplinary engineering.



Christiaan J. J. Paredis
Clemson University, Clemson, USA
paredis@clemson.edu

Christiaan J. J. Paredis is Professor and BMW Endowed Chair in Systems Integration at Clemson University, International Center for Automotive Research (CU-ICAR). His research focuses on model-based systems engineering, combining aspects of decision theory, modeling and simulation, and systems theory to support the design of complex engineered systems. He also directs the Deep Orange education program. From 2014-2016, he served as the Program Director at the National Science Foundation, where he was responsible for programs related to systems engineering and design. Prof. Paredis serves as Associate Editor for the ASME Journal of Computing and Information Science in Engineering, and has received various awards, including the 2011 ASME CIE Excellence in Research Award.

Birgit Vogel-Heuser
Technical University of Munich, Munich, Germany
vogel-heuser@tum.de

Birgit Vogel-Heuser, Prof. Dr.-Ing., is a full professor and director of the Institute of Automation and Information Systems at the Technical University of Munich. Her main research interests are systems engineering, software engineering, and modeling of distributed and reliable embedded systems. She is coordinator of the Collaborative Research Centre (CRC) 768: Managing cycles in innovation processes – integrated development of product-service systems based on technical products, member of acatech, chair of the VDI/VDE working group on industrial agents and vice chair of the IFAC TC 3.1 computers in control.