© 2017 by the authors; licensee RonPub, Lübeck, Germany. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (http://creativecommons.org/licenses/by/4.0/).

Open Access

Open Journal of Information Systems (OJIS)
Volume 4, Issue 1, 2017

www.ronpub.com/ojis
ISSN 2198-9281

Ontology-Based Data Integration in Multi-Disciplinary Engineering Environments: A Review

Fajar J. Ekaputra [A], Marta Sabou [A], Estefanía Serral [B], Elmar Kiesling [A], Stefan Biffl [A]

[A] Institute of Software Technology and Interactive Systems, Technische Universität Wien, 1040 Vienna, Austria,
{fajar.ekaputra, marta.sabou, elmar.kiesling, stefan.biffl}@tuwien.ac.at
[B] Research Center for Management Informatics, KU Leuven, 3000 Leuven, Belgium,
estefania.serralassensio@kuleuven.be

ABSTRACT

Today's industrial production plants are complex mechatronic systems. In the course of the production plant lifecycle, engineers from a variety of disciplines (e.g., mechanics, electronics, automation) need to collaborate in multi-disciplinary settings that are characterized by heterogeneity in terminology, methods, and tools. This collaboration yields a variety of engineering artifacts that need to be linked and integrated, which on the technical level is reflected in the need to integrate heterogeneous data. Semantic Web technologies, in particular ontology-based data integration (OBDI), are promising to tackle this challenge that has attracted strong interest from the engineering research community. This interest has resulted in a growing body of literature that is dispersed across the Semantic Web and Automation System Engineering research communities and has not been systematically reviewed so far. We address this gap with a survey reflecting on OBDI applications in the context of Multi-Disciplinary Engineering Environment (MDEE). To this end, we analyze and compare 23 OBDI applications from both the Semantic Web and the Automation System Engineering research communities. Based on this analysis, we (i) categorize OBDI variants used in MDEE, (ii) identify key problem context characteristics, (iii) compare strengths and limitations of OBDI variants as a function of problem context, and (iv) provide recommendation guidelines for the selection of OBDI variants and technologies for OBDI in MDEE.

TYPE OF PAPER AND KEYWORDS

Research Review: *multi-disciplinary engineering, ontologies, data integration, semantic technologies*

1 INTRODUCTION

The lifecycle of production systems (e.g., manufacturing and power plants) typically involves contributions by engineers from a variety of disciplines [7] that collaborate in *multi-disciplinary engineering environments* (MDEE). For instance, the engineering of a hydro power plant usually involves a main contractor, subcontractors, and component vendors [69]. These stakeholders cover a variety of engineering disciplines (including mechanical, electrical, and automation engineering) and make use of various engineering software tools, datasets, and terminologies, with limited overlap. Collaboration among these stakeholders requires synchronizing and exchanging data produced by software tools specific to their disciplines. In order to overcome the inherent semantic heterogeneity in such settings, data integration is a crucial prerequisite for advanced capabilities to support the work of engineering teams, such as early defect detection [49] or data change management [20]. A key challenge in this context is heterogeneous and semantically overlapping models [22].

Currently, engineers conduct data integration with software tools such as Microsoft Excel and hard-coded data transformers. Integration processes that rely on these tools are typically time-consuming and error-prone [60]. Researchers and practitioners have therefore explored various alternative approaches [14], many of which are based on *Semantic Web* (SW) technologies. SW technologies were originally designed to address data heterogeneity in web-scale settings that pose challenges in terms of data size, heterogeneity, and level of distribution [6]. SW technologies are a family of knowledge-based approaches that rely on formal, shared domain models (i.e., *ontologies* [30]), which enable a broad range of applications [71], such as media publishing and manufacturing design [43].

Ontologies are a key resource for data integration with SW technologies. They capture implicit knowledge across heterogeneous data sources and create semantic interoperability between them [81]. This is known as *ontology-based data integration* (OBDI). In their seminal publication, Wache *et al.* [81] distinguish three OBDI variants, based on what kind of ontologies are used and how these ontologies relate to each other. These variants are (i) the *single-ontology*, (ii) the *multiple-ontology*, and (iii) the *hybrid* OBDI. These variants were identified by studying OBDI system examples from various domains available in 2001.

In recent years, research on OBDI applications for data integration in MDEE has been intensified, e.g., for engineering design quality improvement [22, 33, 74], for simulation generation and evaluation [18, 76], for knowledge representation [52, 63], and for team collaboration [53, 72, 82].

Given the complexity of data integration scenarios in MDEE, choosing the most appropriate OBDI variant, as well as particular suitable technologies is challenging. Appropriate choices are mainly determined by the specific characteristics of the problem setting, such as data source heterogeneity or mapping complexity between the data sources.

The academic literature provides only limited guidance to practitioners in this context, because reports on the use of OBDI in MDEE are dispersed across the SW and *Automation Systems Engineering* (ASE) research communities, which makes it difficult for potential users to gather actionable insights.

To address this challenge, we conducted a literature analysis on OBDI applications in MDEE developed within both the SW and the ASE research communities. The research questions we address in this paper are:

- RQ1: *What key characteristics of data integration scenarios in MDEE affect the choice of an adequate OBDI variant?*
- RQ2: *Which different OBDI variants have been used and what are their strengths and limitations with respect to key characteristics of data integration scenarios in MDEE?*
- RQ3: *What technical alternatives of OBDI elements have been implemented in MDEE?*

Our contributions are relevant for two target groups. Firstly, to **potential users** in the engineering domain and other domains with similar multi-disciplinary characteristics, we provide an overview of OBDI variants, describe their respective characteristics, outline technology options for OBDI in MDEE, and offer a guideline for choosing appropriate OBDI variants and suitable technologies based on characteristics of their data integration scenarios. To this end, we introduce an additional OBDI variant to Wache's typology [81], based on patterns that we observed frequently in OBDI variants in the engineering domain.

Secondly, **to the SW research community**, we provide an overview of research on OBDI applications for data integration scenarios in MDEE from the engineering research community. Further, we report on emerging requirements from the engineering domain that may shape future SW research challenges.

The remainder of this paper is organized as follows: **Section 2** introduces SW technologies and explains the key concepts of MDEE and of OBDI. In **Section 3**, we survey relevant papers categorized according to the production system lifecycle phases they cover. **Section 4** (RQ1) describes key characteristics of data integration scenarios in MDEE. In **Section 5** (RQ2), we identify an additional OBDI variant and compare the strengths and limitations of OBDI variants against a set of MDEE data integration scenario characteristics. In **Section 6** (RQ3), we summarize technology options for OBDI elements in MDEE. In **Section 7**, we discuss our findings and **Section 8** concludes the paper with an outlook for future research.

2 PRELIMINARIES

In this section, we introduce the multi-disciplinary engineering environments (Section 2.1), Semantic Web technologies (Section 2.2), and ontology-based data integration (Section 2.3).

2.1 Multi-Disciplinary Engineering Environment

The VDI 3695 guideline [77] for plant engineering defines an *engineering organization* as a set of engineering teams that is involved in the planning, realization, and commissioning of new technical systems and, if necessary, the optimization or modernization of existing systems.

In this context, an engineering organization becomes the execution environment of a *multi-disciplinary engineering process* that requires collaboration between various engineering disciplines to develop products and the associated production systems [7]. A key characteristic of this execution environment, referred as *multi-disciplinary engineering environments* (MDEE) is the presence of heterogeneous data sources produced by diverse software tools from the involved engineering disciplines [7], where a key challenge consists in obtaining a common view of this data [20]. Current developments in the engineering domain, often associated with the German term "*Industrie 4.0*" [5], require more flexible production systems that rely on strong data integration across various stakeholders and engineering disciplines. Furthermore, the desired shorter lifecycles and higher variation of products in modern production systems requires better integration between (i) the life cycles of products and the associated production systems, and (ii) the engineering and operation phases of these production systems [67].

Consequently, multi-disciplinary engineering processes that create modern and flexible production systems have strong needs for data integration, which must evolve from current, primarily manual practices towards more flexible and knowledge-driven technologies.

2.2 Semantic Web Technologies

The successful implementation of the *World Wide Web* led to an explosive growth of data available on the web [36]. This growth posed challenges for information retrieval and one of the proposed solutions was to annotate web content with machine-processable representations. This idea of applying formal knowledge representation on the web started in the 1990s and was later associated with the vision of a *Semantic Web (SW),* defined by Tim Berners Lee as "*an extension of the current Web, in which information is given well-defined meaning, better enabling computers and people to work in cooperation*" [6]. This well-defined meaning would be established through semantic descriptions, e.g., metadata of web pages.

In order to make these semantic descriptions interpretable by machines and support information retrieval from the web, several principles must be followed [10]. First, semantic descriptions should describe information in terms that impose precise meaning and reflect agreement of a wider community. A collection of these terms and relations between them will form an ontology [30]. Second, semantic descriptions should be expressed in a *representation language* that can be parsed and interpreted by computer programs. In particular, these languages have to have clearly specified semantics that can be leveraged to enable computer programs to derive new information, a process referred to as *inference or reasoning*.

SW technologies were originally developed with the aim to implement the vision of SW [34]. The W3C has published a number of standards for SW technologies that, although they were originally developed for the web, can and have been applied in many other areas, for instance, integration of genome data and media publishing [71][67].

The *Resource Description Framework* (RDF [70]) constitutes the foundation of these standards and provides a graph-based data model. *RDF Schema* (RDFS [31]) provides a lightweight vocabulary description language, whereas the more expressive *Web Ontology Language* (OWL [54]) facilitates specification of rich ontologies. To allow querying, the W3C standardized the *SPARQL protocol and RDF query language* [32]. Furthermore, the SW research community developed technologies, e.g., for acquiring data from various sources[1], mapping between different ontologies (e.g., [50]), improve SPARQL querying performance [25, 29], and reducing the efforts of ontology implementations [9]. Building on and combining these elements, approaches were developed to enable data integration and data access, explained in Section 2.3.

2.3 Ontology-Based Data Integration

Ontology-based data integration (OBDI) refers to the use of (potentially several layers of) ontologies that capture implicit knowledge across heterogeneous data sources to achieve semantic interoperability between these sources [81]. Figure 1(1-3) illustrates three OBDI variants and their components based on a categorization introduced by Wache *et al.*: single-ontology, multiple-ontology, and hybrid OBDI. This classification reflects the number and type of ontologies used for data integration.

[1] https://www.w3.org/wiki/ConverterToRdf

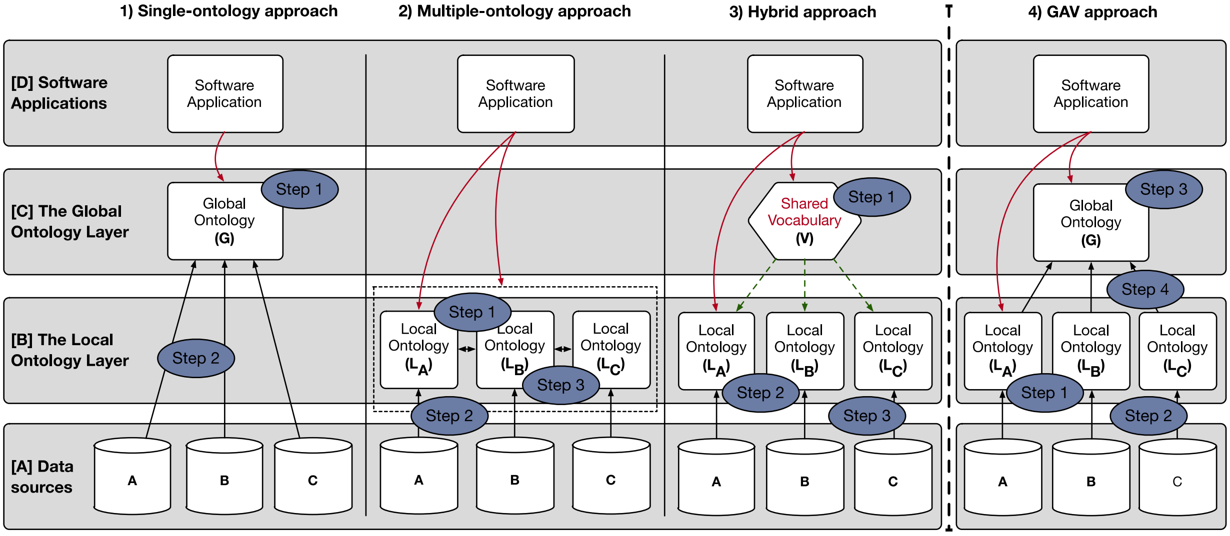

Figure 1: Three variants of OBDI from [75]: (1) single-ontology, (2) multiple-ontology, (3) hybrid, and an additional OBDI variant (4) Global-as-View (GAV).

(Explanation: Red arrows indicate access from an application to data, black arrows represent transformation/virtual access to the data; dotted green arrows represent implicit relations between involved ontologies, and numbered items show the sequence of system development. The dotted rectangle refers to the federation of local ontologies. Section 5.1 explains the additional OBDI variant (4) *Global-as-View* (GAV).)

We distinguish among four layers of OBDI components as shown in Figure 1:

[A] *Data sources* represent the (heterogeneous) local data repositories, which need to be integrated.

[B] The *local ontology layer* contains so-called “local ontologies”, which represent the content of each individual data source repository.

[C] The *global ontology layer* contains so-called “global ontologies”, which are semantically sufficiently broad to represent the data from all data sources to be integrated.

[D] The *software applications layer* represents the applications, which access the data integrated with OBDI.

Assuming three data sources *A*, *B* and *C*, their integration can be achieved by means of three alternative OBDI variants.

1) The **single-ontology** OBDI relies on a single global ontology to integrate all data sources (cf. Figure 1-1). In this approach, the integration process consists of two steps: (i) define a single global ontology *G* and (ii) transform source data from *A*, *B*, and *C* into the global ontology *G*. This integration process is typically hard to maintain because it is susceptible to changes in each data source. Any time a change occurs in one of the data sources, a decision has to be made whether to push the change to the global ontology. If so, to ensure compatibility, the global ontology as well as all mappings to all data sources must be updated.

2) The **multiple-ontology** OBDI involves a local ontology per integrated data source and an alignment of these ontologies with each other using semantic mappings[2] (Figure 1-2). Examples for this mappings include SPIN [48], SPARQL Construct [32], and EDOAL[3]. The integration process consists of three steps: (i) create local ontologies L_A, L_B, and L_C for data sources *A*, *B*, and *C*, respectively, (ii) transform source data of *A*, *B*, and *C* according to their respective local ontologies, and (iii) establish semantic mappings between related ontologies. The drawback of this approach is that semantic mappings among involved ontologies are hard to define and maintain due to varying granularities of the local ontologies. Also, each inclusion of a new data source requires additional semantic mappings to all existing local ontologies.

3) Finally, the **hybrid** OBDI is similar to the multiple-ontology OBDI as it is characterized by definitions of a local ontology per data source. However, instead of independent alignments among local ontologies, this

[2] See [50] for a more comprehensive overview about semantic mapping in the engineering domain.

[3] http://alignapi.gforge.inria.fr/edoal.html

approach defines a shared vocabulary (i.e., a set of basic terms of a domain, which sometimes is also an ontology [78]) to be used and extended within local ontologies, i.e., by means of ontology refinements (see Figure 1-3). In this approach, the integration process consists of three steps: (i) define a shared vocabulary V that contains basic terms/concepts of the domain, (ii) create three local ontologies L_A, L_B and L_C by using and/or extending the shared vocabulary V for data sources A, B, and C respectively, and (iii) transform/annotate source data from A, B, and C according to local ontologies L_A, L_B, and L_C.

Virtual access versus materialization. The ability to integrate data from non-ontology sources is a common requirement in the OBDI context. To this end, *Ontology-Based Data Access* (OBDA) has been developed as a technique to allow virtual data access over data in data sources [13]. In contrast to more traditional approaches such as ETL (i.e., extract, transform, and load), and similar to virtual data access in database schema integration [17], it does not necessarily rely on materialization. In this paper, we consider both virtual access and materialization as implementation options of OBDI approaches that will be discussed further in Section 6.2.

3 A Survey of OBDI Approaches in MDEE

To understand the current landscape of OBDI approaches in the overall lifecycle of production systems, we conducted a literature study and collected a total of 23 OBDI applications from 29 research papers. We explain the survey methodology in Section 3.1 and group the results along plant lifecycle stages [11]:

- **Planning of assembly and production processes.** In this phase, plant planners decide on manufacturing processes and resources necessary for building a plant.
- **Production plant design.** In this phase, engineers work within their respective domains to build the production systems. The phase includes exchange of design data among involved engineering disciplines.
- **Virtual and actual start-up.** The virtual start-up validates the production plant design by systematically iterating through planned and potential plant operation scenarios. The actual start-up of a plant involves plant adjustments on the shop floor after the plant assembly process.
- **Production and service.** Monitoring and improvement of the production plant, manufacturing execution, predictive maintenance, and plant re-configuration are examples of tasks in this phase.

Table 1 summarizes OBDI applications in MDEEs classified along these life-cycle stages. Eleven applications focus on the *design phases* (planning and design) for purposes such as design validation, quality improvement, simulation generation and evaluation (Section 3.2). Six applications focus on the *run-time phases* (startup, production, and service) for system monitoring, diagnostic, evaluation and transient data integration (Section 3.3). The remaining six applications address both design and runtime phases to support tasks such as integrated data analysis (Section 3.4).

3.1 Survey Methodology

We identified relevant research articles from the SW and ASE communities through a systematic literature review (SLR) [47, 84] covering the following steps, described in more detail in the following subsections:

1) **Keyword-based search** on article title published at selected conferences. Different sets of keywords were used for the two target research communities.
2) **Definition of the inclusion/exclusion criteria.** Taking into account the inclusion/exclusion criteria, we analyzed the paper titles/abstracts/content of the retrieved the papers and selected the relevant ones.
3) **Retrieval of further potential articles** from citations and references of selected papers.
4) **Identification of the final set of OBDI applications** from selected papers to be further analyzed.

3.1.1 Keyword-based Search

In our survey, we limit our keyword search to research articles published in five main conferences of the SW community (ISWC, ESWC, i-Semantics/SEMANTiCS, i-KNOW, and EKAW) and three main conferences of the ASE community (ETFA, IFAC, and INDIN) between 2010 and 2016.

SLR Step 1: Keywords-based search. We use a separate set of keywords for SW and ASE conferences. Both sets of keywords omit the "*data integration*" term, as this keyword typically does not appear in the title. For SW conferences, we assume that ontology-related keywords are unnecessary, as it is implied with the article submissions to conferences in this research area. Therefore, we focus on keywords related to the domain, e.g., *engineering* or *production* (cf. Listing 1.A). In contrast, for conferences in the ASE domain, we focus our search on ontology-related keywords with supplementary terms that specify our focus on the domain, which are "*production system*" and "*production plant*" (cf. Listing 1.B).

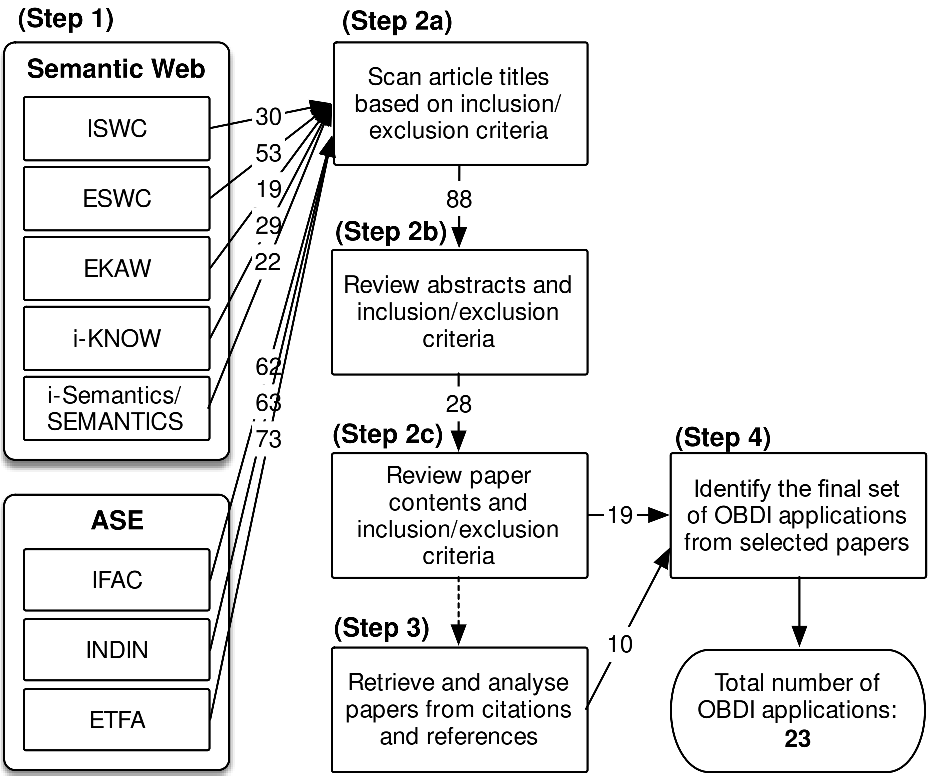

Figure 2: Number of articles and OBDI applications retrieved during literature search

A. Keywords for SW conferences:

```
automation OR engineering OR
product* OR system OR
production OR manufacture* OR
energy OR plant.
```

B. Keywords for ASE conferences:

```
ontology OR semantic OR
knowledge*base OR
'linked data' OR
'production system' OR
'production plant'
```

Listing 1: Keywords used for literature search

We executed the keyword search on all selected conferences between 2010 – 2016 using the *Scopus* search engine[4], with the exception of the 2016 edition of i-KNOW (not indexed by *Scopus – skipped*) and ISWC (metadata did not mention ISWC – *manual search*). The keyword search yielded more than *350 papers* (Figure 2, Step 1).

3.1.2 Definition of Inclusion/Exclusion Criteria

We set the following inclusion and exclusion criteria to remove irrelevant papers from the papers identified with the keyword-based search:

Inclusion Criteria:

- Paper contains scenarios or use cases of data integration using ontologies in the automation system engineering domain.
- Ontology languages or frameworks used for data integration are explicitly mentioned and explained.

Exclusion Criteria:

- The reported approach involves only a single data source.
- Non-OBDA relational database or purely *Eclipse Modelling Framework*[5]-based approaches.

SLR Step 2a: Inclusion/Exclusion. We applied these inclusion and exclusion criteria first on the paper titles, which resulted in a set of 88 papers (Figure 2, Step 2a).

[4] https://www.scopus.com/

[5] http://www.eclipse.org/modeling/emf/

Table 1: An overview of OBDI approaches in MDE (No shading: OBDI variants; with shading: production plant lifecycle phases)

OBDI approach classifications	Single-ontology	Multiple-ontology	Hybrid	Global-as-View	Planning	Design	Start-up	Production & Service
Aarnio *et al.* [1]			X					X
Abele *et al.* [2, 3]	X							X
Brecher *et al.* [11]	X				X	X	X	X
Dibowski &Kabitzsch [15]			X			X		
Dubinin *et al.* [18]				X		X		
Ekaputra *et al.* [20]				X		X		
Feldman *et al.* [22]		X				X	X	
Graube *et al.* (2013) [28]		X				X	X	X
Graube *et al.* (2016) [27]	X							X
Hennig *et al.* [33]	X					X		
Imran and Young [39]				X		X		
Kovalenko *et al.* [49]		X				X		
Lee & Kim [51]	X							X
Lin & Harding [53]				X		X		
Natarajan *et al.* [62]	X							X
Novak and Sindelar [64]	X					X	X	
ONTO-PDM [26, 65]	X							X
Optique [44, 45, 73]		X				X		X
Sabou *et al.* [68]	X					X		
Softic *et al.* [72]	X					X		
Strube *et al.* [74]	X					X		
VFF [42, 75, 76]	X					X	X	X
Wiesner *et al.* [82]			X			X		

SLR Step 2b: Abstract Analysis. Applying the same criteria to the abstracts of the remaining papers, reduced the overall set of papers to 28 papers (Figure 2, Step 2b).

SLR Step 2c: Content analysis. There were cases where the abstract did not clearly justify an article's inclusion or exclusion. In these cases, we analyzed the content of the paper to take the final decision (Figure 2, Step 2c). As a result, we shortlisted 19 papers.

3.1.3 Retrieving Further Potential Articles and Identifying the Final Set of OBDI Applications

SLR Step 3: Retrieval of further potential articles. The keyword-based search only covered a limited number of publications on the topic. To extend our set of considered papers, we conducted an additional search based on references and papers that were cited by the 19 papers from the shortlist we obtained in the previous step. As a result, we added ten additional papers (Figure 2, Step 3) and arrived at set of 29 papers.

SLR Step 4: Identifying the final set of OBDI applications. Some of these papers covered the same approaches or extensions thereof. We group these papers accordingly and arrived at the final 23 OBDI applications (Figure 2, Step 4; Table 1).

3.2 OBDI in the Design Phases

Dibowski and Kabitzsch [15] propose an *Ontology-based Device Descriptions* approach, which aims to provide a formal, unified, and extensible production system device specifications using SW technologies. This approach uses several layers of ontologies, where the top level contains generic domain vocabularies that will be reused and extended in lower layers. Their approach implements a hybrid OBDI, where the top-level ontology is comparable to the shared vocabularies.

Imran and Young [39] demonstrate the potential of formal reference ontologies to support interoperability with a study case of manufacturing bill of materials. They use a Common Logic-based *Knowledge Frame Language* framework to define concepts within assembly systems in a multi-layered ontology approach. Their approach implements a hybrid OBDI with a foundation ontology as shared vocabularies.

Lin and Harding [53] propose using ontologies to support collaboration of engineers involved in a manufacturing system engineering process. The proposal implements a *Global-as View (GAV)* OBDI (see Section 5.1), where the involved organizations develop their independent local ontologies and then map these to the global ontology. These mappings serve as a semantic bridge to exchange and integrate the data across these organizations**.**

Wiesner *et al.* [82] build on their previous work of the *OntoCAPE* ontology [57] to develop an information integration approach in chemical process engineering, which is called the *Comprehensive Information Base* (CIB). CIB adopts the hybrid OBDI where they derive the shared vocabulary from OntoCAPE and develop source (local) ontologies for several local data sources based on the global ontology. They use a two-layer local ontology approach: (i) *Import ontologies*, which are derived directly from data sources (e.g., XML files) using (semi)-automatic data transformation, which are later transformed into (ii) *Document ontologies* that are conformed to the shared vocabularies. They use *F-Logic* instead of the standard RDF/OWL languages to represent all facts, rules, and queries. The authors argue that *F-Logic* is more suitable for defining rules for integration and mapping purposes as well as for the formulation of expressive queries.

Strube *et al.* [74] propose an approach to combine the CAEX data format [38] and SW technologies to support re-developments/modernization of plant automation. The approach involves integrating several CAEX instance files containing plant designs and their proposed changes, together with a set of rule definitions to validate plant changes. These data are integrated using a single-ontology OBDI that is using an adaptation of the CAEX data model as a global ontology. They define a set of SWRL [35] rules for validating the proposed changes in the modernization process of plant automation.

Softic *et al.* [72] semantically integrate data from several data sources to track engineering tasks in an automotive product lifecycle within a single-ontology OBDI. Their architecture consists of three layers: (1) Data layer, where their approach acquires data from local data sources, (2) Entities layer, where they store and link data, and (3) View layer, where users interact with the integrated data. Two different views of data are defined in the view layer: (a) project managers' view and (b) engineers' view, which allow the system to provide different focus on the integrated data.

Dubinin *et al.* [18] introduce an approach based on GAV OBDI for integrating information across data sources in the automation domain. Rather than the typical local ontologies development based on a shared global ontology, they develop the global ontology independent of the local ontologies. To transform local ontology data into instances of the global ontology, they introduce the eSWRL transformation language as an extension of SWRL [35] for RDF-to-RDF transformation.

Kovalenko *et al.* [49] focus on the use of SW technologies to detect defects early in the power plant engineering process. To this end, they adopt the multiple-ontology OBDI to integrate heterogeneous data from several engineering disciplines. They cooperate with domain experts to define links between data from several involved disciplines, i.e., mechanical engineering, electrical engineering, and project management. Furthermore, they develop a set of SPARQL queries to detect defects and validate power plant engineering data.

Ekaputra *et al.* [20] primarily focus on using SW technologies to support data change management within MDEE, where data changes in one engineering discipline need to be validated and propagated to other disciplines. To this end, they adopt a GAV OBDI to represent the heterogeneous data as local and global ontologies. Similar to Dubinin *et al.* [18], they develop both local and global ontologies independently from each other, and they use SPARQL queries to transform, validate and propagate changes between several local ontologies via the global ontology.

Hennig *et al.* [33] propose a SW-based approach to improve the semantic validity and the analysis capability of the multi-disciplinary engineering/system engineering of space systems. To this end, they integrate data from various engineering disciplines within the space system engineering (e.g., mechanical, electrical, instruments, control and software engineering) using the ECSS-E-TM-10-23A data exchange standard as a common (global) data model in a single-ontology OBDI. They focus on the inferencing capability of OWL2 to provide advanced analysis in their scenario.

Sabou *et al.* [68] develop the *AutomationML Analyzer*[6] tool to support engineering of Cyber-Physical Production Systems (CPPS) according to the single-ontology OBDI, where they use an ontology form of the *AutomationML*[7] data exchange format as the global ontology for integrating and analyzing *AutomationML* data from engineering disciplines. The combined data serves as a baseline to provide advanced capabilities to engineers, e.g., analysis and visualization of CPPS engineering design.

3.3 OBDI in the Runtime Phases

Aarnio *et al.* [1] propose an adaptation of a hybrid OBDI to support condition-based monitoring in automation systems. They conduct a four-steps transformation process from local data to RDF:

- Automatic transformation of source data from local source formats to temporary RDF data
- Transformation of temporary RDF data into instances of local ontologies, where the local ontologies conform to shared vocabularies
- Use of the SILK [79] tool to link between data from local ontologies
- Development and execution of rulesets on top of local ontologies to infer new information.

The two-level local ontology approach is similar to the approach in Wiesner *et al.* [82], with the difference that they are using the standard RDF/OWL language to represent both local and global ontologies with the help of SILK. They evaluate their approach with a set of SPARQL queries targeting both local and global ontologies.

Abele *et al.* [3] suggest utilizing SW technologies to support monitoring and diagnostic systems (MDS) in industrial applications. This approach builds on their previous work on the single-ontology OBDI that utilizes the Semantic Media Wiki infrastructure, rule ontology

[6] http://data.ifs.tuwien.ac.at/aml/analyzer/

[7] https://www.automationml.org/

and Drools engine [2]. To this end, they integrate both static plant artifacts data from the design-time engineering and plant component states from run-time engineering to provide users with relevant MDS information.

Graube *et al.* [27] propose a "mixed" solution based on a single-ontology OBDI to integrating static data (e.g., RDF data) and transient data (i.e., sensor data that is coming from web services) based on the URI dereferencing feature of SPARQL 1.1. An evaluation of the proposed solution offers sufficient performance to access transient data as an alternative to the currently available solutions (e.g., SSN, SensorML, and Linked Sensor Middleware).

Lee and Kim propose a framework for engineering collaboration for distributed product development [51]. They use SW technologies to integrate and facilitate the exchange of context information from several data sources (e.g., Bluetooth, PDA, Etc.). To this end, the framework deploys a single-ontology OBDI to model engineering contexts (e.g., locations of users and roles) and uses it to determine relevant services for stakeholders based on context data derived through inference.

Natarajan *et al.* [61] propose an extension of the OntoCAPE ontology [57], which is called OntoSAFE, to provide an application-oriented ontology focused on process supervision in large chemical plants. Later on, they utilize OntoSAFE as a basis for integrating and exchanging complex plant supervision data using the single-ontology OBDI [62].

Kharlamov *et al.* [45] explain the underlying OBDI approach (OPTIQUE) that can be used in MDEE to facilitate data integration using a multiple-ontology OBDI and OBDA. Two example applications in MDEE based on this approach are: Kharlamov [44] and Solomakhina [73]. **Kharlamov *et al.*** [44] propose an OBDA approach to improve access to large, heterogeneous and stream data at a large organization. To support the proposed OBDA approach, they develop a query repository to store both predictive and reactive analysis queries. They evaluate their approach in a large-scale scenario that involves a combination of static data and dynamic data from sensors (> 30GB of new data produced every day). **Solomakhina *et al.*** [73] propose an ontology-based approach to improve the precision and recall of statistical data analysis in the domain of production systems. They integrate data from three different local ontologies that represent power generation facilities (i.e., Turbine, Sensor, and Diagnostics ontologies) with different OWL2 dialects (*OWL2-QL* and *OWL2-DL*). They show that their integration methods, which combine explicit domain models with SW technologies and statistical analysis, yield a better result compared to a pure statistical analysis.

Panetto *et al.* [65] develop an approach to support product data interoperability between applications and stakeholders involved within manufacturing process environments. Their approach implements a single-ontology OBDI, with their proposed ONTO-PDM ontology based on two industry standards (i.e., ISO 10303 [40] and IEC 62264 [37]) as a common data model and mediator between applications during manufacturing process lifecycle. They implement the ontology in both OWL and relational database, and use First Order Logic (FOL) patterns to map between data coming from the two industry standards within the ONTO-PDM ontology. **Giovannini *et al.*** [26] extend ONTO-PDM with concepts and rules on sustainability principles and technology knowledge. In addition, the authors propose a knowledge base system that use formalized knowledge for supporting product design and process planning. The approach uses SWRL rules to infer additional information and conduct analysis related to sustainability of products.

3.4 OBDI in the Overall Plant Lifecycle

Brecher *et al.* [11] aim for software tool integration in production plant lifecycles with SW technologies. Their approach implements the single-ontology OBDI. They develop an information model as a common ontology for production plant lifecycles and connect a set of software tools via data interpreter and generic interfaces. They use the Globally Unique Identifiers or unique names to identify the same objects in different data sources. The integrated data is used to navigate through production plant lifecycles, including the planning phase of the production process and the assembly process.

Feldmann *et al.* [22] introduce an inconsistency management approach based on SW technologies. The approach integrates two types of data: *SysML4Mechatronics* data that represent the mechatronics architecture and *Matlab/Simulink* data representing workpieces throughput of the plant in a system that implements a multiple-ontology OBDI. In this approach, relations between the two ontologies are defined manually by domain experts. They develop a set of SPARQL queries to detect inconsistencies in the integrated data and successfully retrieve inconsistency of the data as intended in their evaluation.

Graube *et al.* [28] suggest using linked data to allow orchestration of software applications in the production system environment. Their approach implements a multiple-ontology OBDI, where they represent various data sources (e.g., device details, plant structure, report-and-form information, and live data access) as separate local ontologies, and store the information about and the relation among these ontologies in a separate ONT ontology. These ontologies are then orchestrated to build various applications (e.g., Task-List applications and Neighborhood-Browser for data flow explorations) related to production systems.

Novak and Sindelar [64] proposes a single-ontology OBDI to support simulation design and integration of simulation models in industrial automation. The authors develop the automation ontology that serves as the global ontology of the approach that is wrapped in a java-based tool. The tool receives input data from engineers (plant designs) as well as knowledge about devices in the particular industrial plant and available simulation libraries. As outputs, it produces executable simulation configuration files for simulators based on SPARQL query result on automation ontology instances.

Kádár *et al.* [42] propose the *Virtual Factory Framework (VFF)*, an integrated collaborative environment to support the design and management of factory entities. VFF initiate a global ontology (Virtual Factory Data Model - VFDM) for integrating and representing factory objects related to production systems, resources, processes, and products, resembling the *single-ontology OBDI*. A Virtual Factory Manager builds on top of the VFDM to manage and provide access to the VFDM data from various connected tools. These tools act both as data providers as well as data users. **Terkaj and Urgo** [76] focuses on integrating static data of production systems and their performance history, builds on their previously explained VFF. The method allows evaluation of a system design by simulating its performance based on system and simulation logs. **Terkaj *et al.*** [75] extends VFF to evaluate the impact of planning and maintenance decisions during the operation phase of a manufacturing system. They report on an application case of roll-shop system designs, where they develop a graphical user interface and combine it with a Discreet Event Simulation tool to evaluate the performance of roll-shop system configurations.

4 CHARACTERISTICS OF DATA INTEGRATION SCENARIOS IN MDEE

As discussed in Section 2.1, MDEEs are characterized by the involvement of engineers from various engineering disciplines. This collaboration results in the need for integrating heterogeneous data sources produced by domain-specific software tools. We discuss characteristics of data integration scenarios in MDEE that we identified and generalized in our survey to address *RQ1: What key characteristics of data integration scenarios in MDEE affect the choice of an adequate OBDI variant?*

Identifying these characteristics is also the first step to establish criteria that practitioners can use to choose appropriate OBDI variants for their settings.

4.1 Data Integration Objectives

There is a wide range of objectives for data integration in multi-disciplinary engineering settings. In this paper, we do not directly derive recommendations for OBDI variant selection based on these objectives, but focus on the relationships between setting characteristics – explained in Section 4.2 and 4.3 – and OBDI variants. The data integration objectives we compiled from the papers are as follows (summarized in Table 2):

Table 2: Data Integration Objectives for OBDI in MDEE
(No shading: data-related objectives; light shading: overall-system objectives; dark shading: collaboration objectives)

OBDI objectives	**Data Change-Management**	**Transient Data Integration**	**Centralized Engineering Repository**	**Integrated Data Analysis**	**Design Quality Improvement**	**Design Validation**	**Simulation Generation & Evaluation**	**System Monitoring, Diagnostic, & Evaluation**	**Team Collaboration**	**Software Interoperability**
Aarnio *et al.* [1]				X				X		
Abele *et al.* [2, 3]								X		
Brecher *et al.* [11]			X	X						
Dibowski & Kabitzsch [15]				X						X
Dubinin *et al.* [18]							X			
Ekaputra *et al.* [20]	X			X					X	
Feldman *et al.* [22]				X	X					
Graube *et al.* [28]										X
Graube *et al.* [27]		X	X							
Hennig *et al.* [33]			X		X	X				
Imran and Young [39]										
Kovalenko *et al.* [49]				X	X					
Lee & Kim [51]										X
Lin & Harding [53]			X						X	
Natarajan *et al.* [62]			X					X		
Novak & Sindelar [64]							X			
ONTO-PDM [26, 65]									X	X
Optique [44, 45, 73]		X	X	X						
Sabou *et al.* [68]			X	X						
Softic *et al.* [72]									X	
Strube *et al.* [74]					X	X				
VFF [42, 75, 76]			X			X	X	X		X
Wiesner *et al.* [82]			X						X	

Objectives related to data:

- **Data Change Management** refers to the process of managing local data changes and their effects on the overall system [20]. For this particular scenario, data integration serves as a foundation to enable data change management.
- **Transient Data Integration**, such as stream data integration, aims to integrate transient data sources with combination with non-transient data [27, 44].
- **Centralized Engineering Repository** data integration scenarios aim to provide a centralized engineering repository [11, 27, 33, 44, 53, 62, 68, 76, 82].
- **Integrated Data Analysis** refers to typical OBDI approaches that aim to enable data analysis on top of integrated OBDI data [1, 11, 20, 22, 44, 49, 68, 73].

Objectives related to the overall system:

- **Design Quality Improvement** aims at improving the quality of system design in MDEE, e.g., with inconsistency management [22] or defect detections [33, 49, 74] over a global view of data sources.
- **Design Validation** aims to validate system designs against a set of validation criteria based on integrated data [33, 42, 74, 76].
- **Simulation Generation and Evaluation** aim to generate [18, 76]) and evaluate [76] system simulation in MDEE.
- **System Monitoring, Diagnostic and Evaluation** aim for system monitoring [1, 3, 62], diagnostic [3] and evaluation [42] in MDEE.

Objectives related to collaborations:

- **Team Collaboration**. This goal refers to the use of integrated data for supporting team collaborations [20, 53, 72, 82].
- **Software Interoperability**. This goal aims to provide a "common language" for software partners to interact with each other (e.g., for app orchestration [28], intelligent service finder [51], or data exchange [42]).

4.2 Data Sources

In this section, we explain data-source related characteristics of MDEE scenarios.

Data types. The primary focus of a multi-disciplinary engineering process is on the structured data. Spreadsheets, XML-based data formats, RDF, streaming/sensor data, and relational databases are the most common data types in the MDEE as shown in Table 3.

Several scenarios also report the use of specific data formats, e.g., AutomationML[8] for data exchange, SysML[9] for plant design, and ECSS-E-TM-10-23A[10] for space engineering.

Number of data sources. Due to our focus on OBDI approaches in research communities, data integration scenarios typically report on the integration of a small number (i.e., less than ten) of data sources.

Size of data. There is a large variety in the size of data, ranging from cases with the least amount of tens of data points [20] up to those that can handle more than 30 GB of sensor data daily [44].

Table 3: Data source types for OBDI in MDEE

OBDI data source types	Relational Database	Spreadsheets	XML-Based	RDF	Streaming / Sensor Data	Others / Specific Data Formats
Aarnio *et al.* [1]	X	X		X		
Abele *et al.* [2, 3]					X	
Brecher *et al.* [11]						X
Dibowski & Kabitzsch [15]						X
Dubinin *et al.* [18]				X		
Ekaputra *et al.* [20]		X				
Feldman *et al.* [22]				X		X
Graube *et al.* (2013) [28]	X				X	
Graube *et al.* (2016) [27]					X	
Hennig *et al.* [33]						X
Imran and Young [39]						X
Kovalenko *et al.* [49]			X	X		
Lee & Kim [51]					X	
Lin & Harding [53]				X		
Natarajan *et al.* [62]					X	
Novak and Sindelar [64]	X					X
ONTO-PDM [26, 65]						X
Optique [44, 45, 73]	X				X	
Sabou *et al.* [68]			X			
Softic *et al.* [72]			X			
Strube *et al.* [74]			X			
VFF [42, 75, 76]	X					X
Wiesner *et al.* [82]			X			

[8] https://www.automationml.org
[9] http://www.omgsysml.org/
[10] http://ecss.nl/hbstms/ecss-e-tm-10-23a-space-system-data-repository/

Data source dynamics. The addition and removal of data sources can be crucial for engineering scenarios. Several engineering scenarios consider this data source dynamics [1, 82], while others do not.

Data access. Most scenarios need access to the integrated data as a whole. Some scenarios, however, report on the requirement to access both local *and* global (parts of integrated) data for various reasons, e.g., to compare local data from different sources [1, 22, 82] or to enable data change propagation [20].

4.3 Semantic Heterogeneity

Semantic heterogeneity reflects differences between two or more data sources. The heterogeneity in data integration systems varies between individual cases in MDEE. As an example, the semantic heterogeneity is small in data integration cases where engineers develop most of their local data sources according to a data standard (e.g., *AutomationML* [68], CAEX [74], OPC-UA [27] and ECSS-E-TM-10-23A [33]).

However, there are cases where local data source structures are created independently without prior agreement or standard as a basis (e.g., hydropower plant UC [20, 49]). In these cases, we cannot assume any prior agreement among data owners and must rely on mapping definitions of source structures (or between data sources and *common data structure*, depending on the chosen data integration approach) to enable interoperability. In these cases, the semantic heterogeneity is in general considerably higher.

Mapping complexity reflecting the complexity of relations among involved data sources varies across scenarios. This characteristic is important due to the differences of OBDI variant capabilities to represent mappings.

5 Survey Result Analysis

In this section, we analyze the survey result in relation with OBDI variants and MDEE data integration scenario characteristics. We propose an extension of the OBDI classification by Wache *et al.* in Section 5.1 and the analysis in Section 5.2. We conclude with a summary table in Section 5.3.

5.1 Global-as-View OBDI

Looking back into OBDI categorizations from Wache *et al.* [81] (cf. Figure 1-1, 1-2, and 1-3), we observe that there are OBDI applications that are similar (i.e., they make use of a global ontology and several local ontologies), but do not exhibit all the characteristics of hybrid OBDI [18, 20, 53]. Specifically, these applications develop local ontologies *before* the definition of the global ontology (cf. Figure 1-4). Therefore, the local and global ontologies are independent from each other. In this situation, interoperability is achieved by transforming local ontologies into instances of a global ontology. We refer to this approach as **Global-as-View (GAV) OBDI** due to its similarity (i.e., it contains a global schema without modifying local schemas) with the GAV approach from the relational databases [17]. To differentiate this OBDI approach from existing variants, we propose to add GAV to the typology (cf. Figure 1-4).

GAV OBDI requires the definition of one local ontology per data source, similar to multiple-ontology and hybrid OBDI. In this method, the integration process consists of four steps (cf. Figure 1-4): (i) Creation of three independent local ontologies L_A, L_B, and L_C (or reuse of existing local ontologies) for data sources *A*, *B*, and *C* respectively. (ii) Transformation of source data in local sources *A*, *B*, and *C* according to local ontologies L_A, L_B, and L_C. (iii) Development of a global ontology *G* represents a set of common concepts relevant to scenarios, and (iv) Definition of independent mappings between a local repository (i.e., L_A, L_B, and L_C) and the global ontology *G* to facilitate data transformation from local ontologies to the global ontology.

Several researchers, e.g., Gagnon [24], Modoni *et al.* [55], and Moser [58, 59] have proposed ideas similar to, or having common points with the GAV OBDI without differentiating it to existing OBDI variants, while Juarez *et al.* report an adoption of GAV OBDI in a related domain of home automation [41]. In this paper, we formulate and differentiate GAV from other OBDI variants.

5.2 OBDI Variants Analysis based on the MDEE Characteristics

In this section, we evaluate each OBDI variant (i.e., *single-ontology, multiple-ontology, hybrid and GAV OBDI*) against a set of MDEE scenario characteristics from Section 4 (i.e., *semantic heterogeneity, data access, mapping complexity, and data source dynamic*). Furthermore, we consider *ontology implementation effort* as an additional criterion.

5.2.1 Single-ontology OBDI

Single-ontology OBDI is common in MDEE – more than half of the papers surveyed belong to this category.

Semantic heterogeneity. Single-ontology OBDI is convenient when data sources are semantically close [3, 11, 42, 44, 51, 62, 72, 76] or when they can be aligned according to a common data standard (e.g., *AutomationML* [68], CAEX [74], OPC-UA [27] and ECSS-E-TM-10-23A [33]).

Data access. Software applications built on top of a single-ontology OBDI infrastructure can only access the global ontology, i.e., they cannot access data that are not captured in the global ontology.

Mapping complexity. Because only a single (global) ontology is used, single-ontology OBDI typically does not require any mapping definitions. In some cases, where semi-automatic global ontology acquisition is possible (e.g., [68]), mappings are needed to transform intermediate ontology instances (i.e., the automatically generated local ontologies from data sources) according to the global ontology.

Data source dynamics. Changes to the global ontology are costly, also because they may affect transformation mechanisms from local ontologies. Therefore, the single-OBDI approach is more suitable for scenarios with infrequent data source additions or if addition of a data source does not affect the global ontology.

Ontology implementation effort. Single-ontology OBDI requires only the development of a global ontology, but no additional inter-ontology mappings.

5.2.2 Multiple-ontology OBDI

Semantic heterogeneity. Each data source is described independently using a local ontology, without an implicit assumption that these local ontologies share vocabularies. Therefore, multiple-ontology OBDI is suitable in scenarios with high semantic heterogeneity.

Data access. Each local ontology can be accessed independently, an aggregation of local ontologies can be made accessible using named graphs [22, 49] or an aggregated ontology [28, 44, 73] can be used. In principle, the aggregated local ontology could also be accessed via SPARQL *Federated Queries* [66], although we did not encounter an implementation of it in the survey.

Mapping complexity. Multiple-ontology OBDI requires a set of mappings that define relations among the involved local ontologies. We found that most applications of multiple-ontology OBDI ([23, 45, 49, 73]) use RDF property mappings to represent these relationships. There is only one exception [28] that uses instance mappings instead.

Data source dynamics. Each addition of a new data source to a multi-ontology OBDI infrastructure requires (i) the definition of new local ontology and (ii) mappings from the new local ontology to other local ontologies. This implies that adding data sources involves considerable effort. Most implementations in our survey involve a fixed number of data sources and a limited number of mappings and do not consider data source dynamics. Graube *et al.* [28] hint at the possibility of adding new data sources, but the authors do not explain how their application would address such dynamics.

Ontology implementation effort. The approach requires development of a set of local ontologies and the definition of a set of mappings among them. This is acceptable for scenarios with a limited number of local sources and mappings, which were common in our survey [23, 45, 49, 73]. For more complex cases, however, alternative OBDI approaches are necessary.

5.2.3 Hybrid OBDI

Semantic heterogeneity. A central concept in hybrid OBDI is the availability of a shared vocabulary that facilitates the integration of data sources, not only those that have a similar view of a domain (i.e., low semantic heterogeneity), but also those with a high level of semantic heterogeneity.

Data access. Hybrid OBDI provides two ways to access data: (i) direct access to the (aligned/restructured) local ontologies, and (ii) access to the shared vocabulary, where the system queries each local ontology and merges the results. Aarnio *et al.* [1] demonstrate and evaluate both access methods, and they report that direct access to local ontologies is faster than access to the shared vocabulary. Wiesner *et al.* [82] focus more on accessing the integrated data via shared vocabularies.

Mapping complexity. Hybrid OBDI defines mappings between local and global ontologies using semantic relations. To this end, this approach typically uses a set of RDF properties as reported in [1] (e.g., `owl:sameAs` and `owl:subClassOf`). In applications that do not rely on SW technologies (but rather, e.g., *F-Logic* [82]), authors typically do not report on how relationships among involved ontologies are established.

Data source dynamics. Hybrid OBDI makes integration of additional data sources easier through the shared vocabulary refinement method. Reports on hybrid OBDI [1, 82] hint at this capability without discussing it in detail or considering dynamics in their application.

Ontology implementation effort. Initial development of a hybrid OBDI system involves considerable effort. Stakeholders need to reach an agreement on the definition of shared vocabularies and need to develop (or redesign, if local ontologies are already available) local ontologies for each data source based on the shared vocabulary. However, these efforts then result in aligned local ontologies without need for additional mappings.

Table 4. Characteristics, strengths and limitations of OBDI variants
(Green: strengths, yellow: slight limitations; red: limitations)

	Single-ontology	Multiple-ontology	Hybrid	GAV OBDI
Semantic Heterogeneity	best applied for data sources similar view of a domain	support heterogeneous views	support heterogeneous views	supports heterogeneous views
Data Access	only allows access on global data	allows access on each (original, if any) local ontology and the aggregated local ontologies.	allows access on each (restructured) local ontology and the global ontology.	allows access on each (original, if any) local ontology and the global ontology
Data Source Dynamics (*addition of data sources*)	needs for some adaptation in the global ontology	needs to provide a new local ontology and map the new local ontology to other local ontologies	only needs to provide (or restructure) local ontology based on the shared vocabulary	needs to provide a new local ontology and define mappings to the global ontology
Mapping Complexity	N/A	supports simple mappings (semantic relations)	supports simple mappings (vocabulary refinement)	supports simple and complex mappings (queries and rules)
Ontology Implementation Effort	straightforward	costly	reasonable	rather costly

5.2.4 Global-as-View OBDI

Semantic heterogeneity. Similar to the hybrid OBDI approach, the availability of a "common view" of a global ontology in Global-as-View (GAV) OBDI can address various levels of heterogeneity.

Data access. GAV OBDI provides access on the global and local ontology levels. In line with this capability, MDEE data integration scenarios using GAV OBDI provide access to both local and global ontologies [18, 20, 53].

Mapping complexity. Mappings between local and global ontologies are represented by a set of transformation rules or queries. Depending on the scenario, the mappings can be one-way (local-to-global, e.g., [18, 53]) or two-ways (local-to-global and global-to-local [20]), with various levels of complexity.

Data source dynamics. GAV OBDI requires several steps to include an additional data source. First, it is necessary to define or reuse a local ontology for the new data source. Then, transformation rules to the global ontology have to be established. It does not, however, require other local ontologies and mappings to change. Two reports [20, 53] highlight this as an advantage of the approach.

Ontology implementation effort. The effort required to establish the ontologies and their mapping is comparable to the effort for hybrid OBDI, albeit with a different use of such mapping (i.e., for transforming instead of linking RDF data instances). SPARQL *Construct* [20], eSWRL (an extension of SWRL rule language) [18], and arbitrary transformation code [53] are example languages that are used for this kind of transformation. *TopBraid* SPIN[11] can also serve as an alternative, however, so far none of the approaches has been used in an MDEE.

5.3 Summary of OBDI Characteristics

Table 4 summarizes comparison results and highlights the strengths and limitations of OBDI variants in MDEE based on the analysis in Section 5.2.

Wache *et al.* [81] consider hybrid OBDI the most effective variant. We observe, however, that single-ontology OBDI is the most popular OBDI approach in MDEE due to its simplicity (i.e., it is suitable for scenarios where there is no need to preserve local data structures). If users need to keep local data source structures and compare instances from these sources, other OBDI variants are more suitable.

6 TECHNICAL REALIZATION OF OBDI ELEMENTS

This section explains technical realization options for OBDI elements. We focus our investigation to the OBDI elements shown in Figure 3, including (i) *Ontology Language and Framework*, (ii) *Data Acquisition*, (iii) *Mapping and Transformation*, and (iv) *Storage and OBDI data access*. We report on the results of our survey for each of these elements in Sections 0 - 6.4.

[11] http://spinrdf.org/

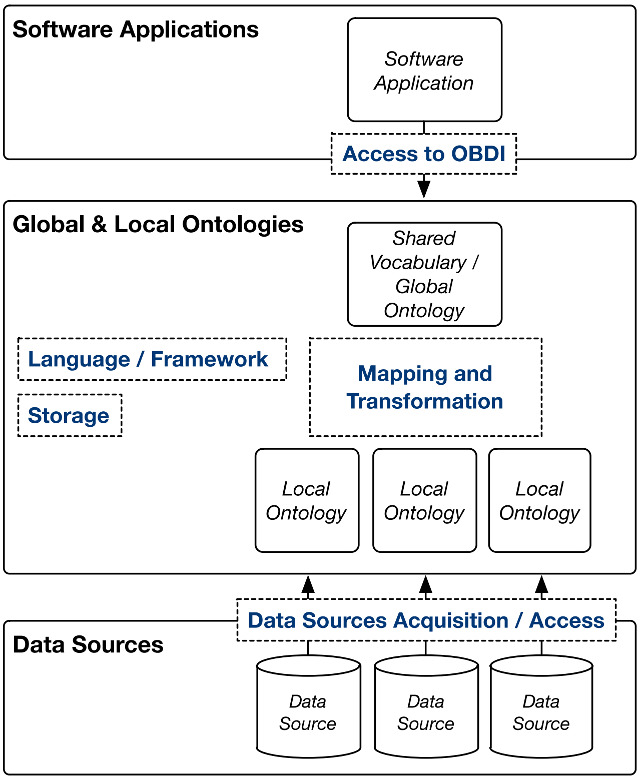

Figure 3: OBDI solution elements (blue texts)

6.1 Ontology Language and Framework

In recent years, the *Resource Description Framework* (**RDF**) for expressing information about resources [32], together with **RDF Schema** as a data modeling vocabulary [31], and *Web Ontology Language* (**OWL**) as an ontology language [80] emerged as the *de facto* standard for representing ontologies on the *Semantic Web* (SW). Most SW-based OBDI applications use these three standards. Abele *et al.* [3] propose an alternative approach on top of the RDF-based **Semantic Media Wiki**[12]. Several of these approaches use standard and custom RDF vocabularies, e.g., SSN and DUL to represent sensor data [73], IEC-61499 ontology [18], *SysML* and *Matlab/Simulink* ontologies [22].

A few of the surveyed approaches do not use W3C standard-based ontology languages/frameworks. Wiesner *et al.* [82] rely on **F-Logic** [46] to represent all facts, rules, and queries. They argue that even though the combination of OWL and the rule language SWRL [35] can in principle provide the same level of expressiveness as *F-Logic*, it has drawbacks, e.g., the lack of negations. *F-Logic* could define rules for integration and mapping purposes as well as formulations of expressive queries. Imran and Young [39] use similar arguments for their selection of **Common Logic**-based *Knowledge Frame Language* (KFL) and emphasize that KFL is more expressive and has more powerful reasoning capabilities compared to OWL. Lee and Kim [51] use XML **Topic Maps**[13], which were proposed as an alternative to RDF at the time of their research. Because W3C standards are the dominant approach, the following sections will focus on the RDF(S) and OWL.

6.2 Data Acquisition

OBDI approaches in the engineering domain typically integrate structured data in various formats. Most approaches in our survey integrate relational databases [1, 28, 42, 44, 73, 76], spreadsheets [1, 20], XML [49, 68, 72, 74, 82], and RDF graph data [1, 18, 22, 49, 53]. Several OBDI approaches also integrate specific or legacy data formats, e.g., *SysML* [22], CAEX [74], web services and ECAD [11], and ECSS-E-TM-10-23A [33].

Several approaches are possible to integrate non-ontology data into an ontology graph. The *Extract, Transform, and Load* (**ETL**) mechanism is one of the most used, where OBDI approaches develop custom

[12] https://www.semantic-mediawiki.org/

[13] http://www.topicmaps.org/

applications to convert data (e.g., [42, 76], [15], [26, 65]).

The *Extract, Load, and Transform* (**ELT**) mechanism represents another method, which may involve automatic conversion to an ontology graph (e.g., [1, 20, 49, 82]). This mechanism first transforms data source instances to an arbitrary ontology graph and then transforms the resulting graph into a target ontology representation. In comparison to ETL, ELT transforms data *within* a single ontology language.

The *Ontology-Based Data Access* (**OBDA**) method allows users to access virtual RDF graphs of non-RDF data source instances, mainly from relational databases (e.g., [44, 73] use *Ontop* [12]). RML Mapping Language [16] facilitates OBDA for other data sources (e.g., XML, JSON, and CSV). However, we have not found an RML application in approaches within our survey.

Graube *et al.* [27] propose a method to **acquire transient data** (e.g., web services that contain sensor data) as part of their OBDI implementation. They adapt the URI dereferencing functionality of *SPARQL 1.1 Service Description* [83] to retrieve web services data during SPARQL query executions.

Due to the preliminary nature and the small amount of data involved, Hennig *et al.* [33], Dibowski and Kabitzsch [15], and ONTO-PDM [26, 65] used **manual** data acquisition/transformation of source data to RDF.

6.3 Semantic Mapping and Transformation

We observe that most OBDI approaches in our survey rely on either a single or one method of the following combinations of methods for mapping definitions: RDF property mapping, Globally Unique Identifier (GUID) matching, a combination of both RDF property mapping and GUID matching, or property value matching.

- **RDF property mapping** relies on a set of RDF properties to link classes, properties and instances of different ontologies, e.g., `owl:sameAs`, `rdfs:subClassOf`, `rdfs:subPropertyOf`, `owl:equivalentClass` and custom RDF properties [22, 44, 49, 53],[1, 73].
- **URI/GUID matching** links instances of ontologies with identical URIs [11, 20, 27, 28, 33, 68, 74, 82], [1, 73], [15], [64]. The approach rests on the assumption that individuals will be assigned a unique identifier across different local ontologies in the acquisition process.
- **Property value matching** is another method used for instances mapping, where two or more objects in different ontologies are considered the same if certain property values of these instances are the same [18], [26, 65].

To define these mappings and create the actual relations, OBDI applications employ RDF to RDF transformation methods and tools, such as **SILK** [79] (e.g., [1]), **SPARQL construct** queries (e.g., [20]) and **arbitrary transformation code** based on RDF APIs (e.g., [27]). Within these tools, algorithms for finding links among these ontologies are deployed, e.g., string matching or custom user-defined rules.

An alternative to the transformation methods and tools are **reasoners and rule engines**. We found a number of them in our survey, e.g., Wiesner *et al.* [82] use the *OntoBroker* [4] rule engine to define rules for mapping, Natarajan *et al.* [62] use the *Hermit* reasoner[14] to improve the querying process, and ONTO-PDM [26, 65] use *first order logic* (FOL) to define instance relations based on property values. Hennig *et al.* [33] use Pellet[15] and Strube *et al.* [74] use SWRL with *Jess*[16] to derive implicit knowledge.

6.4 Storage and OBDI Data Access

In our survey, we identify three RDF-based storage options: RDF *triplestore*, in-memory store and relational databases. Wiesner *et al.* use the **OntoBroker** storage system for their *F-Logic* based ontologies.

- **RDF triple stores** (e.g., *Virtuoso*[17], *Jena* TDB[18], *StarDog*[19] or RDF4J[20]) allow users to store large RDF data as triples [1, 22, 49, 68, 72, 76], [15], [26, 65]. Cf. [56] for a comparison of selected RDF store solutions in MDEE.
- The **in-memory store** [20, 53], [39], [64] is often used for smaller-scale data, e.g., for prototypes or proof-of-concepts.
- The use of **relational databases** via an OBDA layer are also common [44, 73]. Despite efforts from the SW community, the capabilities of RDF triple-stores are still lacking behind relational databases. Relational databases with an OBDA layer are often used in scenarios that need to cope with large amounts of data.

[14] http://www.hermit-reasoner.com/
[15] https://github.com/stardog-union/pellet
[16] http://www.jessrules.com/
[17] https://virtuoso.openlinksw.com/
[18] https://jena.apache.org/documentation/tdb/
[19] http://www.stardog.com/
[20] http://rdf4j.org/

Table 5. Technology options for OBDI elements and their adoptions in MDEE
("X" indicates adoption; "-" indicate that no clear information available in the paper)

OBDI Approach in MDEE	***OBDI Variant***				***Language and Framework***							***Data Acquisition***					***Mapping***			***Transformation***				***Data Storage***				***Data Access***			
	Single-ontology	Multiple-ontology	Hybrid	GAV OBDI	RDF	OWL	OWL2	F-Logic	Topic Maps	SMW	Common Logic (CL)	ETL	ELT	OBDA	Transient Data Acquisition	Manual	RDF Property	URI/GUID Matching	Property Value Matching	SILK	SPARQL Construct	Arbitrary Transformation Code	Reasoner & Rule Engine	Triplestore	In-memory / file-based	Relational DB	Others (e.g., ontobroker)	SPARQL Endpoints	Custom APIs	Custom GUIs	Stream Data Engine
Aarnio *et al.* [1]			X		X	X							X				X	X		X				X				X			
Abele *et al.* [2, 3]	X				X					X		X						X		-				X		X		X	X		
Brecher *et al.* [11]	X				X	X						-						X				X		-					X		
Dibowski and Kabitzsch [15]			X		X		X					X						X				X		X				X			
Dubinin *et al.* [18]				X	X	X						-							X				X		X			-			
Ekaputra *et al.* [20]				X	X	X							X					X			X				X				X		
Feldman *et al.* [22]		X			X	X						X					X					X		X				X			
Graube *et al.* (2013) [28]		X			X	X						X						X				X		X				X			
Graube *et al.* (2016) [27]	X				X	X									X			X				X		X				X			X
Hennig *et al.* [33]	X				X		X									X		X				X	X		X			-			
Imran and Young [39]				X							X					X	-					X			X				X		
Kovalenko *et al.* [49]		X			X	X							X				X					X		X				X			
Lee & Kim [51]	X								X			X					-					X		-					X		
Lin & Harding [53]				X	X	X											X					X			X			-			
Natarajan *et al.* [62]	X				X		X					-					-						X	-						X	
Novak and Sindelar [64]	X					X						X						X				X			X				X		
ONTO-PDM [26, 65]	X				X		X									X			X				X	X				X			
Optique [44, 45, 73]		X			X		X							X			X			-						X		X		X	
Sabou *et al.* [68]	X				X	X							X					X				X		X				X			
Softic *et al.* [72]	X				X	X						X					-			-				X						X	
Strube *et al.* [74]	X				X	X							X					X					X	-					X		
VFF [42, 75, 76]	X				X	X						X					-					X		X					X	X	
Wiesner *et al.* [82]			X					X					X					X					X				X		X		

The three most widely used mechanisms to access OBDI data from software applications are **SPARQL endpoints** (e.g., [1, 22, 28, 49, 68], [15], [26, 65]), **API-based services** (e.g., [20, 42, 74, 76, 82], [39], [64]), and **custom-build GUI**s (e.g., [62, 72, 75]). Furthermore, extensions of SPARQL endpoints are being developed to allow access to streaming data [27].

Table 5 summarizes technology options used as part of OBDI approaches on papers within our survey.

7 DISCUSSION

In this section, we discuss our findings for each of the three investigated research questions.

RQ1: *What key characteristics of data integration scenarios in MDEE affect the choice of an adequate OBDI variant?*

Based on our survey, we identified key characteristics for data integration in MDEE (see the detailed explanations in Section 4). We selected these characteristics based on the following criteria: (i) relevancy to the MDEE domain, which is reflected in the occurrence of these characteristics in published papers, (ii) effects on the choice of different OBDI variants, and (iii) their variance across engineering scenarios (e.g., we do not consider the number of data sources because it is similar across all the papers surveyed). The selected characteristics are:

- *Semantic heterogeneity* refers to the degree to which the structure of local data sources differs. Semantic heterogeneity varies across OBDI scenarios.
- *OBDI data access* refers to the expected type of access to OBDI data, e.g., several scenarios only require access to global ontology data to perform their analysis [68], while other cases need to access both local ontologies and the global ontology for performing data change management [20].
- *Data source dynamics* captures whether the addition of data sources is considered important or necessary in the scenario.
- *Mapping complexity* characterizes the complexity of relations that can be established between ontologies. Simple mappings can be represented using RDF properties, whereas more complex relations may occur in MDEE scenarios that require other means of representation.

RQ2: *Which different OBDI variants have been used and what are their strengths and limitations with respect to key characteristics of data integration scenarios in MDEE?*

Single-ontology OBDI. In this OBDI variant, the shared vocabulary of all the data sources that need to be integrated is defined in a single global ontology. Data from various data sources are transformed into instances of the global ontology to achieve the data integration.

The approach is convenient to use when various data sources are semantically close or when data sources can be transformed into a "*common language*" of the domain (e.g., AutomationML). If such semantic closeness or a "*common language*" are not available, any addition or removal of data sources may require adaptation of the global ontology to avoid loss of information. Our survey revealed, however, that this approach appears to be sufficient for most MDEE scenarios: more than half of the studied cases adopt this approach. We assume that this popularity is due to the low implementation effort it requires (e.g., only one ontology needs to be built, no ontology mapping/alignment is required).

Multiple-ontology OBDI. Each data source in a multiple-ontology OBDI is described using its local ontology. We cannot assume that these local ontologies share any joint vocabulary. Mappings are established between the local ontologies.

The advantage of this approach is that there is no commitment among local ontologies to shared vocabularies or a global ontology; however, this is also the most significant disadvantage due to the difficulties of relating content in different local ontologies. To overcome this drawback, inter-ontology mappings between local ontologies have to be added. However, these mapping definitions become more difficult when more data sources are being introduced to the system, since local ontologies have to be mapped to each other, which constitutes an exponential problem. The multiple-ontology OBDI is hence more suitable for scenarios where there are a limited number of data sources and therefore a manageable number of inter-ontology mappings is needed. For more complex data integration scenarios, other OBDI variants are more appropriate.

Hybrid OBDI is characterized by the availability of a shared vocabulary that contains basic terms of a domain that local ontologies should build on via vocabulary/ontology refinement.

The shared vocabulary allows linking and comparing instances from multiple local ontologies, which are relevant for multiple data integration scenarios in multidisciplinary environments. This approach reduces the effort required to define inter-ontology mappings among local ontologies. However, this approach has its drawback: it forces re-development of local ontologies – including their mappings to local data sources – in order to comply with the shared vocabulary. As such, the

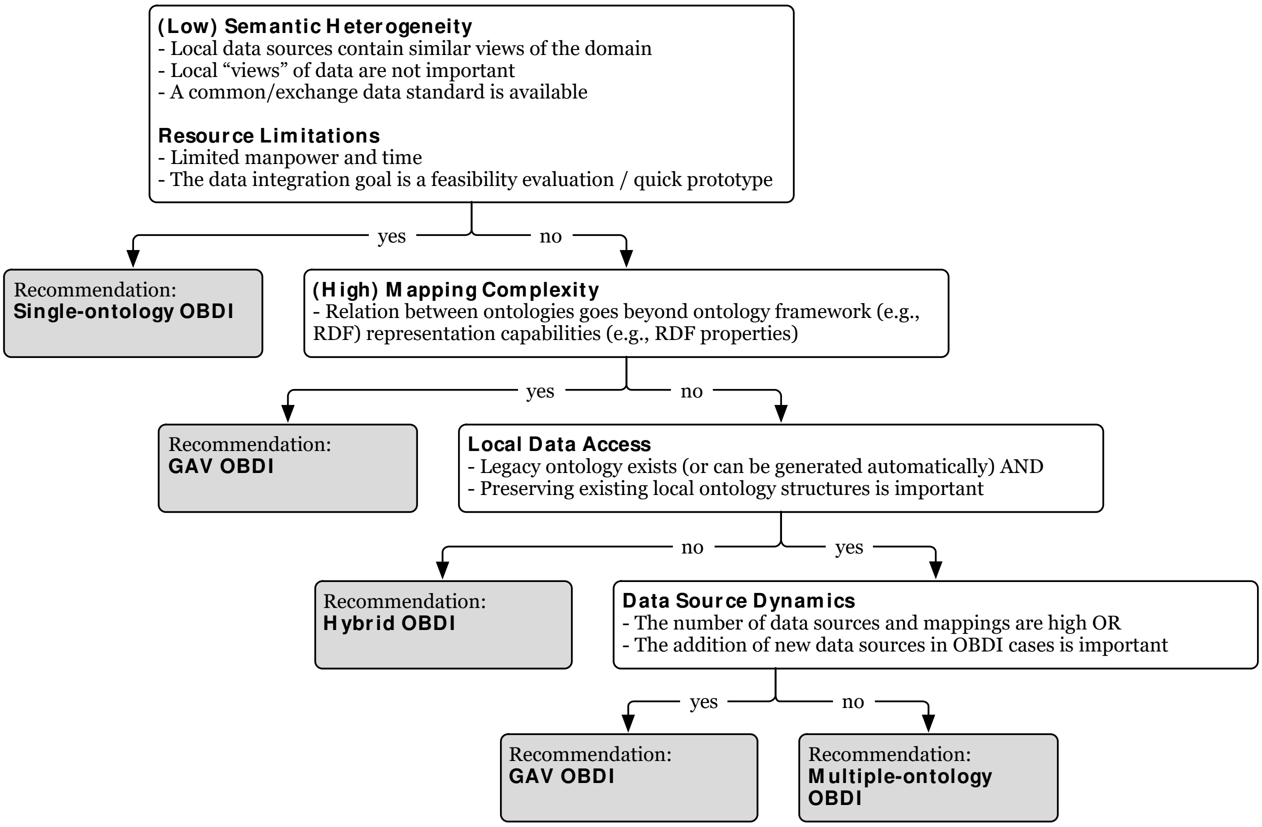

Figure 4 The OBDI approach recommendation tree

hybrid OBDI is less suitable for MDEE cases where local ontologies are already established (e.g., in a brownfield OBDI scenario) or they can be automatically generated from data sources.

Global-as-View (GAV) OBDI. The central concept of the GAV approach lies in the global ontology definition, which is similar to the hybrid OBDI. GAV OBDI, however, does not require re-development of existing local ontologies due to inter-ontology transformation definitions between local and global ontologies similar to those used in the multiple-ontology OBDI. In this way, existing local ontologies can be preserved and mapping definitions can be added to allow comparison among local ontologies. Furthermore, data sources can be added with moderate effort (i.e., mappings between the local ontology representing the new data source and the global ontology). Additionally more complex relations beyond ontology representation capabilities are possible (e.g., to represent complex engineering mappings from [50]).

OBDI Recommendation Tree. We developed the OBDI approach recommendation tree (**Figure 4**) based on the OBDI characteristics (cf. Table 4) in MDEE scenarios, the OBDI comparison table by Wache *et al.* [81], and our analysis result in Section 5.2. The tree summarizes our discussion of RQ2 considering several factors (i.e., *semantic heterogeneity, resource limitations, mapping complexity, local data access/preservation, and data source dynamics*) and can serve as a guideline for practitioners and researchers in selecting the most suitable OBDI approach for the characteristics of their scenario.

RQ3: *What technical alternatives of OBDI elements have been implemented in MDEE?*

In Section 6, we report on a set of technical realizations of OBDI elements in MDEE. We categorize our observations into the following four groups:

- **Language and Framework**. Most OBDI adoptions in MDEE use the RDF framework for their implementation. Two alternatives are reported: Topic Maps and *F-Logic*. The main reason of using *F-Logic* is its capability to accommodate data and rules (e.g., constraints, custom inferences). OWL on its own does not have such a capability. This capability can only be achieved with additional rule languages, such as SWRL [82] or the *Jena* rule language. At present, however, there are no W3C recommendations to define such rules.

- **Data Acquisition**. We identified three mainstream approaches for acquiring data from local sources, namely ETL, ELT, and OBDA. We also found a unique (and potential) approach for data acquisition that tries to integrate transient data with a SPARQL extension [27] as well as manual acquisition [33].
- **Mapping and Transformation**. To define mappings between ontology classes and individuals, URI matching and RDF properties (either from standard, e.g., `owl:sameAs`, or custom RDF properties) are mainly used. In order to achieve this, transformation methods and tools are used. In some cases, manual mapping and transformation processes are conducted due to the limited number of data that does not warrant the effort of developing dedicated automated mapping methods.
- **Data Storage and OBDI Data Access**. In several scenarios, the native RDF triple-store is not sufficient [44, 73], and therefore, relational databases with OBDA appear to be the only viable alternative. Hybrid storage solutions combining elements of the traditional approach (i.e., a relational database) and SW solutions [33] have also been proposed. We also explained several data access methods for OBDI data, namely SPARQL endpoints, custom APIs and GUIs and SPARQL extensions (see Section 6.4 for more details).

Threats to Validity:

As with every empirical study, there are threats to validity that may introduce bias and, therefore, need to be considered and addressed. For this study, we see the following most relevant threats to validity and the countermeasures we took.

Selection of literature sources. The survey may miss important papers outside of the selected scope. As a countermeasure, we chose a comprehensive scope and include the major conferences in the SW and ASE research communities. We expect these conferences to be representative of the target research communities. In addition, we went through citations listed in the papers we identified in our structured survey and included additional relevant papers outside of these conferences.

Researcher bias. Researcher bias may be introduced by personal bias and oversight, particularly if only a single researcher is involved. To mitigate the risk of researcher bias, we followed a well-structured standard process [47, 84] and involved several researchers in each stage in order to achieve a balanced view.

Limited information on technical OBDI elements. Another limitation we found was the unavailability of data on some technical aspects for several approaches (see Table 5). However, we consider the current set of collected data as representative with respect to the overall target scope of OBDI applications in MDEE.

8 CONCLUSIONS

In this paper, we report on a review of *Ontology-Based Data Integration* (OBDI) approaches in *multi-disciplinary engineering environments* (MDEEs). Our survey covers both the *Semantic Web* (SW) and *Automation Systems Engineering* (ASE) research communities.

Based on the papers identified in a systematic literature review, we derived a set of data integration characteristics in the MDEE domain, proposed an extension to the classification of OBDI conceptual approaches, and evaluated the suitability of different OBDI variants against the derived characteristics. Our proposed classification will be useful not only in the multi-disciplinary engineering domain, but also in other domains with similar characteristics, e.g., scholarly data [8, 21].

Furthermore, we identified an additional OBDI variant not considered in prior categorizations, the so-called *Global-as-View* (GAV) ontology approach. We differentiate the GAV from other OBDI approaches and discuss the strengths and limitations of various OBDI variants. One of the main advantages of the GAV approach is its ability to preserve existing local ontology structures for analysis purposes.

We observed technology options for OBDI elements from the selected papers. We find that most of their implementations are using W3C standards of SW technologies (i.e., RDF-based approachs). There are, however, several approaches using alternative technologies, due to their maturity for industrial uptake, e.g., *F-Logic* as an alternative of RDF [82]. We also observed feedback from the engineering community with regards to their adoption of SW technologies in their domain, e.g., inadequate storage performance [33], high-learning curve [49], and the unavailability of rules and transformation standards [18, 20].

Directions for future work include **extending our survey beyond the engineering domain** to verify and generalize our findings w.r.t OBDI scenario characteristics, conceptual classifications, and their adoptions. In this future survey, additional criteria (e.g., ontology reuse and publishing) and quantitative comparisons (e.g., the number of mappings and ontologies) can be added.

Another line of future work is research on the **expressiveness of the ontology framework**. Several OBDI approaches use non-SW ontology frameworks (i.e., KFL and *F-Logic*), arguing that SW technologies are not sufficiently expressive for their data integration needs. However, the limitations of SW technologies have not been systematically investigated in this context.

Finally, we also plan to build upon knowledge gathered in this survey to develop **an OBDI-based data change management** approach to improve the effectiveness of multi-disciplinary engineering processes by reducing the amount of necessary manual work [19].

ACKNOWLEDGEMENTS

This work was supported by the *Christian Doppler Forschungsgesellschaft* (CDG), the Federal Ministry of Economy, and Science, the *Österreichischer Austausch-dienst* (ÖAD), the *Österreichische Forschungs-förderungsgesellschaft* (FFG) grant 861213 (CitySPIN), and the *Linked Data Lab* at TU Wien.

REFERENCES

[1] P. Aarnio, I. Seilonen, and M. Friman: "Semantic repository for case-based reasoning in CBM services", in *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*, pp. 1–8, 2014.

[2] L. Abele and S. Grimm: "Knowledge-based integration of industrial plant models", in *IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society*, pp. 4392–4397, 2013.

[3] L. Abele, S. Grimm, S. Zillner, and M. Kleinsteuber: "An ontology-based approach for decentralized monitoring and diagnostics", in *2014 12th IEEE International Conference on Industrial Informatics*, pp. 706–712, 2014.

[4] J. Angele, M. Kifer, and G. Lausen: "Ontologies in F-Logic", in *Handbook on Ontologies*, S. Staab and R. Studer, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 45–70, 2009.

[5] T. Bauernhansl, M. ten Hompel, and B. Vogel-Heuser: *Industrie 4.0 in Produktion, Automatisierung und Logistik: Anwendung-Technologien-Migration*. 2014.

[6] T. Berners-Lee, J. Hendler, and O. Lassila: "The semantic web", *Scientific American*, vol. 284, no. 5, pp. 34–43, 2001.

[7] S. Biffl, D. Gerhard, and A. Lüder: *Multi-disciplinary engineering for cyber-physical production systems*. Springer International Publishing, 2017.

[8] S. Biffl, M. Kalinowski, F. J. Ekaputra, E. Serral, and D. Winkler: "Building empirical software engineering bodies of knowledge with systematic knowledge engineering", in *Proceedings of the 26th International Conference on Software Engineering and Knowledge Engineering*, pp. 13–3, 2014.

[9] M. Blasko, P. Kremen, and Z. Kouba: "Ontology evolution using ontology templates", *Open Journal of Semantic Web (OJSW)*, vol. 2, no. 1, pp. 16–29, 2015. [Online] http://nbn-resolving.de/urn:nbn:de:101:1-201705194898

[10] M. Boškovi, E. Bagheri, D. GaŠevi, B. Mohabbati, N. Kaviani, and M. Hatala: "Automated staged configuration with semantic web technologies", *International Journal of Software Engineering and Knowledge Engineering*, vol. 20, no. 4, 2010.

[11] C. Brecher, D. Özdemir, J. Feng, W. Herfs, K. Fayzullin, M. Hamadou, and A. W. W. Müller: "Integration of software tools with heterogeneous data structures in production plant lifecycles", in *10th IFAC Workshop on Intelligent Manufacturing Systems*, pp. 48–53, 2010.

[12] D. Calvanese, B. Cogrel, S. Komla-Ebri, R. Kontchakov, D. Lanti, M. Rezk, M. Rodriguez-Muro, and G. Xiao: "Ontop: Answering SPARQL queries over relational databases", *Semantic Web*, vol. 8, no. 3, pp. 471–487, 2016.

[13] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, A. Poggi, M. Rodriguez-Muro, R. Rosati, M. Ruzzi, and D. F. Savo: "The MASTRO system for ontology-based data access", *Semantic Web*, vol. 2, no. 1, pp. 43–53, 2011.

[14] Y. Chen: "Industrial information integration - A literature review 2006–2015", *Journal of Industrial Information Integration*, vol. 2, pp. 30–64, 2016.

[15] H. Dibowski and K. Kabitzsch: "Ontology-based device descriptions and triple store based device repository for automation devices", in *15th IEEE Conference on Emerging Technologies & Factory Automation*, pp. 1–9, 2010.

[16] A. Dimou, M. Vander Sande, P. Colpaert, R. Verborgh, E. Mannens, and R. Van de Walle: "RML: A generic language for integrated RDF mappings of heterogeneous data", in *the 7th Workshop on Linked Data on the Web*, 2014.

[17] A. Doan, A. Halevy, and Z. Ives: *Principles of*

Data Integration. Elsevier, 2012.

[18] V. Dubinin, V. Vyatkin, C.-W. Yang, and C. Pang: "Automatic generation of automation applications based on ontology transformations", in *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation*, pp. 1–4, 2014.

[19] F. J. Ekaputra: "Knowledge Change management and analysis in engineering", in *Semantic Web Technologies for Intelligent Engineering Applications*, S. Biffl and M. Sabou, Eds. Cham: Springer International Publishing, 2016, pp. 159–178.

[20] F. J. Ekaputra, M. Sabou, E. Serral, and S. Biffl: "Knowledge change management and analysis during the engineering of cyber physical production systems", in *Proceedings of the 12th International Conference on Semantic Systems*, pp. 105–112, 2016.

[21] F. J. Ekaputra, M. Sabou, E. Serral, and S. Biffl: "Supporting information sharing for reuse and analysis of scientific research publication data", in *the 4th Workshop on Semantic Publishing,* 2014.

[22] S. Feldmann, S. J. I. Herzig and K. Kernschmidt, T. Wolfenstetter, D. Kammerl et al.: "Towards effective management of inconsistencies in model-based engineering of automated production systems", in *15th IFAC Symposium on Information Control Problems in Manufacturing*, pp. 916–923, 2015.

[23] S. Feldmann, K. Kernschmidt and B. Vogel-heuser: "Combining a SysML-based modeling approach and semantic technologies for automatic validation of manufacturing plant models", *Procedia CIRP*, vol. 17, pp. 451–456, 2014.

[24] M. Gagnon: "Ontology-based integration of data sources", in *Proceedings of the 10th International Conference on Information Fusion*, pp. 1–8, 2007.

[25] J. M. Gimenez-Garcia, J. D. Fernandez, and M. Martinez-Prieto: "MapReduce-based solutions for scalable SPARQL querying", *Open Journal of Semantic Web (OJSW)*, vol. 1, no. 1, pp. 1–18, 2014. [Online] http://nbn-resolving.de/urn:nbn:de:101:1-201705194824

[26] A. Giovannini, A. Aubry, H. Panetto, M. Dassisti, and H. El Haouzi: "Knowledge-Based System for Manufacturing Sustainability", *IFAC Proceedings Volumes*, vol. 45, no. 6, pp. 1333–1338, 2012.

[27] M. Graube, L. Urbas, and J. Hladik: "Integrating industrial middleware in Linked Data collaboration networks", in *Proceedings of 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 1–8, 2016.

[28] M. Graube, J. Ziegler, L. Urbas, and J. Hladik: "Linked data as enabler for mobile applications for complex tasks in industrial settings", in *Proceedings of the 18th IEEE Conference on Emerging Technologies & Factory Automation (ETFA)*, pp. 1–8, 2013.

[29] S. Groppe, D. Heinrich, and S. Werner: "Distributed join approaches for W3C-conform SPARQL endpoints", *Open Journal of Semantic Web*, vol. 2, no. 1, pp. 30–52, 2015. [Online] http://nbn-resolving.de/urn:nbn:de:101:1-201705194910

[30] T. R. Gruber: "A translation approach to portable ontology specifications", *Knowledge Acquisition*, vol. 5, no. 2, pp. 199–220, 1993.

[31] R. V Guha and D. Brickley: "RDF Schema 1.1", W3C Recommendation, 2014. [Online] https://www.w3.org/TR/rdf-schema/.

[32] S. Harris and A. Seaborne: "SPARQL 1.1 overview", W3C Recommendation, 2013, [Online] http://www.w3.org/TR/sparql11-query/.

[33] C. Hennig, A. Viehl, B. Kämpgen, and H. Eisenmann: "Ontology-based design of space systems", in *Proceedings of the Semantic Web Conference*, pp. 33–48, 2016.

[34] P. Hitzler, M. Krötzsch, S. Rudolph, M. Krotzsch, and S. Rudolph: *Foundations of Semantic Web Technologies*. CRC Press, 2009.

[35] I. Horrocks, P. F. Patel-schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean: "SWRL : A semantic web rule language combining OWL and RuleML", W3C Submission, 2004. [Online]: https://www.w3.org/Submission/SWRL/.

[36] B. Huberman and L. Adamic: "Internet: Growth dynamics of the World-Wide Web", *Nature*, vol. 401, no. 6749, pp. 131–131, 1999.

[37] IEC: "IEC 62264 - International standard for enterprise-control system integration", IEC Standard, 2003. [Online] https://www.iso.org/standard/57308.html.

[38] IEC: "IEC 62424 - Representation of process control engineering-requests in P&I diagrams and data exchange between P&ID tools and PCE-CAE tools", IEC Standard, 2008. [Online]

http://standards.globalspec.com/std/1104045/iec-62424.

[39] M. Imran and R. I. M. Young: "Reference ontologies for interoperability across multiple assembly systems", *International Journal of Production Research*, vol. 54, no. 18, pp. 5381–5403, 2016.

[40] International Organization for Standardization: "ISO 10303 - Standard for the exchange of product model data", ISO Standard, 1999. [Online] https://www.nist.gov/document-2058.

[41] J. Juarez, J. A. Rodriguez-Mondejar, R. Garcia-Castro, J. Juárez, J. A. Rodríguez-Mondéjar, and R. García-Castro: "An ontology-driven communication architecture for spontaneous interoperability in home automation systems", in *Proceedings of the IEEE Emerging Technology and Factory Automation*, pp. 1–4, 2014.

[42] B. Kádár, W. Terkaj, and M. Sacco: "Semantic virtual factory supporting interoperable modelling and evaluation of production systems", *CIRP Annals - Manufacturing Technology*, vol. 62, no. 1, pp. 443–446, 2013.

[43] S. El Kadiri, W. Terkaj, E. N. Urwin, C. Palmer, D. Kiritsis, and R. Young: "Ontology in engineering applications", in *Proceeding of 7th International Workshop of Formal Ontologies Meet Industry*, pp. 126–137, 2015.

[44] E. Kharlamov, N. Solomakhina, Ö. L. Özçep, D. Zheleznyakov, T. Hubauer, S. Lamparter, M. Roshchin, A. Soylu, and S. Watson: "How semantic technologies can enhance data access at siemens energy", in *Proceedings* of *13th International Semantic Web Conference,* pp. 601–619, 2014.

[45] E. Kharlamov, E. Jiménez-Ruiz, D. Zheleznyakov, D. Bilidas, , M. Giese, *et al.*: "Optique: Towards OBDA systems for industry", in *Proceedings of ESWC Satellite Events*, pp. 125–140, 2013.

[46] M. Kifer and G. Lausen: "F-logic: a higher-order language for reasoning about objects, inheritance, and scheme", in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 134–146, 1989.

[47] B. Kitchenham and S. Charters: "Guidelines for performing systematic literature review in software engineering", Keele University Technical Report - EBSE-2007-01, 2007.

[48] H. Knublauch, J. A. Hendler, and K. Idehen: "SPIN - Overview and motivation", W3C Submission, 2011. [Online] https://www.w3.org/Submission/spin-overview/.

[49] O. Kovalenko, E. Serral, M. Sabou, F. J. Ekaputra, D. Winkler, and S. Biffl: "Automating cross-disciplinary defect detection in multi-disciplinary engineering environments", in *Proceedings* of *International Conference on Knowledge Engineering and Knowledge Management*, vol. 8876, pp. 238–249, 2014.

[50] O. Kovalenko and J. Euzenat: "Semantic matching of engineering data structures", in *Proceedings of Semantic Web Technologies for Intelligent Engineering Applications*, pp. 137–157, 2016.

[51] J. Y. Lee and K. Kim: "A distributed product development architecture for engineering collaborations across ubiquitous virtual enterprises", *The International Journal of Advanced Manufacturing Technology*, vol. 33, no. 1–2, pp. 59–70, 2007.

[52] Y. Liao, M. Lezoche, H. Panetto, N. Boudjlida, and E. R. Loures: "Semantic annotation for knowledge explicitation in a product lifecycle management context: A survey", *Computers in Industry*, vol. 71, pp. 24–34, 2015.

[53] H. K. K. Lin and J. A. A. Harding: "A manufacturing system engineering ontology model on the semantic web for inter-enterprise collaboration", *Computers in Industry*, vol. 58, no. 5, pp. 428–437, 2007.

[54] D. L. McGuinness and F. van Harmelen: "OWL Web Ontology Language Overview", W3C Recommendation, 2004. [Online] https://www.w3.org/TR/owl-features/.

[55] G. E. Modoni, M. Doukas, W. Terkaj, M. Sacco, and D. Mourtzis: "Enhancing factory data integration through the development of an ontology: From the reference models reuse to the semantic conversion of the legacy models", *International Journal of Computer Integrated Manufacturing*, vol. 30, no. 10, pp. 1043–1059, 2017.

[56] G. E. Modoni, M. Sacco, and W. Terkaj: "A survey of RDF store solutions", in *Proceedings of International Conference on Engineering, Technology and Innovation*, 2014.

[57] J. Morbach, A. Wiesner, and W. Marquardt: "OntoCAPE - A (re)usable ontology for computer-aided process engineering", *Computers & Chemical Engineering*, vol. 33, no. 10, pp. 1546–1556, 2009.

[58] T. Moser: "Semantic integration of engineering environments using an engineering knowledge

base", Vienna University of Technology, 2009.

[59] T. Moser: "The engineering knowledge base approach", in *Semantic Web for Intelligent Engineering Applications*, S. Biffl and M. Sabou, Eds. Vienna, Austria: Springer, 2016.

[60] T. Moser and S. Biffl: "Semantic Integration of Software and Systems Engineering Environments", *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 1, pp. 38–50, 2012.

[61] S. Natarajan, K. Ghosh, and R. Srinivasan: "An ontology for distributed process supervision of large-scale chemical plants", *Computers & Chemical Engineering*, vol. 46, pp. 124–140, 2012.

[62] S. Natarajan and R. Srinivasan: "Implementation of multi agents based system for process supervision in large-scale chemical plants", *Computers and Chemical Engineering*, vol. 60, pp. 182–196, 2014.

[63] E. Negri, L. Fumagalli, M. Garetti, and L. Tanca: "Requirements and languages for the semantic representation of manufacturing systems", *Computers in Industry*, vol. 81, pp. 55–66, 2016.

[64] P. Novak and R. Sindelar: "Ontology-based industrial plant description supporting simulation model design and maintenance", in *the 39th Annual Conference of the IEEE Industrial Electronics Society*, 2013.

[65] H. Panetto, M. Dassisti, and A. Tursi: "ONTO-PDM: Product-driven ONTOlogy for product data management interoperability within manufacturing process environment", *Advanced Engineering Informatics*, vol. 26, no. 2, pp. 334–348, 2012.

[66] E. Prud'hommeaux and C. Buil-Aranda: "SPARQL 1.1 federated query", W3C Recommendation, 2013. [Online] https://www.w3.org/TR/sparql11-federated-query/.

[67] M. Sabou, F. J. Ekaputra, and S. Biffl: "Semantic web technologies for data integration in multi-disciplinary engineering", in *Multi-Disciplinary Engineering for Cyber-Physical Production Systems*, 2017.

[68] M. Sabou, F. Ekaputra, O. Kovalenko, and S. Biffl: "Supporting the engineering of cyber-physical production systems with the AutomationML analyzer", in *Proceedings of 1st International Workshop on Cyber-Physical Production Systems*, pp. 1–8, 2016.

[69] M. Sabou, O. Kovalenko, F. J. Ekaputra, and S. Biffl: "Semantic web solutions in engineering", in *Semantic Web Technologies for Intelligent Engineering Applications*, S. Biffl and M. Sabou, Eds. Cham: Springer International Publishing, pp. 281–296, 2016.

[70] G. Schreiber and Y. Raimond: "RDF 1.1 primer", W3C Recommendation, 2014. [Online] https://www.w3.org/TR/rdf11-primer.

[71] N. Shadbolt, T. Berners-Lee, and W. Hall: "The semantic web revisited", *IEEE Intelligent Systems*, vol. 21, no. 3, pp. 96–101, 2006.

[72] S. Softic, M. Rosenberger, A. Denger, J. Fritz, and A. Stocker: "Semantically based visual tracking of engineering tasks in automotive product lifecycle", in *Proceedings of the 13th International Conference on Knowledge Management and Knowledge Technologies*, pp. 1–4, 2013.

[73] N. Solomakhina, T. Hubauer, S. Lamparter, M. Roshchin, and S. Grimm: "Extending statistical data quality improvement with explicit domain models", in *Proceedings of the 12th IEEE International Conference on Industrial Informatics (INDIN)*, pp. 720–725, 2014.

[74] M. Strube, S. Runde, H. Figalist, and A. Fay: "Risk minimization in modernization projects of plant automation - A knowledge-based approach by means of semantic web technologies", in *Proceedings of the IEEE Emerging Technology and Factory Automation*, pp. 1–8, 2011.

[75] W. Terkaj, T. Tolio, and M. Urgo: "A virtual factory approach for in situ simulation to support production and maintenance planning", *CIRP Annals - Manufacturing Technology*, vol. 64, no. 1, pp. 451–454, 2015.

[76] W. Terkaj and M. Urgo: "Ontology-based modeling of production systems for design and performance evaluation", in Proceedings of *12th IEEE International Conference on Industrial Informatics*, pp. 748–753, 2014.

[77] Verein Deutscher Ingenieure: "VDI Richtlinie 3695 - Engineering von Anlagen - Evaluieren und optimieren des Engineerings", VDI Standard, 2009. [Online] https://www.vdi.de/richtlinie/vdivde_3695_blatt_2-engineering_von_anlagen_evaluieren_und_optimieren_des_engineerings_themenfeld_prozesse/.

[78] U. Visser, H. Stuckenschmidt, H. Wache, and T. Vögele: "Enabling technologies for interoperability", In *Proceedings of the 14th International Symposium of Computer Science*

for Environmental Protection, pp. 35–46, 2000.

[79] J. Volz, C. Bizer, M. Gaedke, and G. Kobilarov: "Silk - A Link Discovery Framework for the Web of Data", in *2nd Workshop of Linked Data on the Web*, 2009.

[80] W3C OWL Working Group: "OWL 2 web ontology language document overview", W3C Recommendation, 2009. [Online] http://www.w3.org/TR/owl2-overview/.

[81] H. Wache, T. Voegele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Hübner: "Ontology-based integration of information-a survey of existing approaches", in *Workshop on Ontologies and Information Sharing*, pp. 108–117, 2001.

[82] A. Wiesner, J. Morbach, and W. Marquardt: "Information integration in chemical process engineering based on semantic technologies", *Computers & Chemical Engineering*, vol. 35, no. 4, pp. 692–708, 2011.

[83] G. Williams: "SPARQL 1.1 service description", W3C Recommendation, 2013. [Online] http://www.w3.org/TR/2013/REC-sparql11-service-description-20130321/.

[84] A. Zaveri, A. Rula, A. Maurino, and R. Pietrobon: "Quality assessment for linked data: A survey", *Semantic Web*, vol. 7, no. 1, pp. 63–93, 2016.

AUTHOR BIOGRAPHIES

Fajar J. Ekaputra is a PhD candidate and junior researcher at TU Wien. His research topic focuses on Semantic Web Technologies for data integration and knowledge change management in the context of multi-disciplinary environment settings. During his time in TU Wien, he has taken part in designing and implementing research use cases for industry partners in several domains. He co-authored 4 book chapters and more than 15 papers in peer-reviewed venues, winning two best paper awards. Fajar is actively involved in research communities as a scientific reviewer for several international conferences and workshops, e.g., ISWC, SEMANTiCS, IFAC, and BigScholar.

Dr. Marta Sabou is a Senior Researcher at the TU Wien. Prior to this, she was an Assistant Professor at the MODUL University Vienna and a Research Fellow at the Open University. She holds a PhD from VU Amsterdam, for which she won the IEEE Intelligent System's Ten to Watch Award in 2006. Dr. Sabou has performed much of her work in the Semantic Web community where she investigated research topics ranging from ontology engineering to the creation of intelligent systems that benefit from semantic technologies in domains as varied as tourism, open government and industrial automation in the Industrie 4.0 context. Her current interests ar in the use of Human Computation technologies to solve knowledge acquisition challenges in Semantic Web, Natural Language Processing and Software Engineering. She acts as an editorial board member for three journals that publish Semantic Web research and has been engaged in several senior organization activities in the main Semantic Web conferences.

Dr. Estefanía Serral is a senior researcher with a highly international and inter-disciplinary profile. Currently working at KU Leuven (Belgium), she is doing research in topics such as business process management, ubiquitous business processes, and context-adaptive systems. From 2012 to 2014, she led the Semantic Knowledge Representation and Integration research group at the CDL-Lab in the TU Wien (Austria). Before, she worked in the ProS Research Center at the Technical University of Valencia (Spain), where she designed a novel method for developing ubiquitous systems using Model-Driven Development (MDD) and Semantic technologies. Dr. Serral has many publications in high-ranking conferences and journals, such as CAiSE, ER, UIC, PMC, ESWA, SOSYM, MTAP, etc. She completed her PhD in Computer science in 2011; a Master Degree on Software Engineering, Formal Methods and Information Systems in 2008; and a bachelor degree in Computer science in 2006.

Elmar Kiesling, PhD joined TU Wien as a postdoctoral researcher in 2012 after completing his PhD at the University of Vienna. He has contributed numerous scientific publications in a wide range of research areas including Linked Data, operations research and management science, decision analysis, agent-based simulation, innovation, and blended learning.

Prof. Stefan Biffl is an associate professor of software engineering at the Institute of Software Technology and Interactive Systems, TU Wien. He has been the head of the Christian Doppler research laboratory "Software Engineering Integration for Flexible Automation Systems", the scientific program chair of Software Quality Days, and co-author of more than 150 peer-reviewed publications including books on "Value-Based Software Engineering" and "Best-Practice Software Engineering". His research interests and experience include (1) Software quality management, process improvement, and value-based Software Engineering, (2) Software quality assurance, defect detection, inspection, and defect prediction models, (3) Empirical software engineering, evaluation, data collection and integration for analysis, (4) Engineering environment and knowledge integration, and (5) Application area: software and systems engineering companies, business software companies.