A Knowledge Graph Based Integration Approach for Industry 4.0

Dissertation
zur
Erlangung des Doktorgrades (Dr. rer. nat.)
der
Mathematisch-Naturwissenschaftlichen Fakultät
der
Rheinischen Friedrich-Wilhelms-Universität Bonn

von
Irlán Grangel-González
aus
Holguin, Kuba

Bonn, 14.02.2019

Dieser Forschungsbericht wurde als Dissertation von der Mathematisch-Naturwissenschaftlichen Fakultät der Universität Bonn angenommen und ist auf dem Hochschulschriftenserver der ULB Bonn http://hss.ulb.uni-bonn.de/diss_online elektronisch publiziert.

1. Gutachter:	Prof. Dr. Sören Auer
2. Gutachter:	Prof. Dr. Jens Lehmann

Tag der Promotion:	14.02.2019
Erscheinungsjahr:	2019

Abstract

The *fourth industrial revolution*, Industry 4.0 (I40) aims at creating smart factories employing among others Cyber-Physical Systems (CPS), Internet of Things (IoT) and Artificial Intelligence (AI). Realizing smart factories according to the I40 vision requires intelligent human-to-machine and machine-to-machine *communication*. To achieve this communication, CPS along with their data need to be described and interoperability conflicts arising from various representations need to be resolved. For establishing interoperability, industry communities have created standards and standardization frameworks. Standards describe main properties of entities, systems, and processes, as well as interactions among them. Standardization frameworks classify, align, and integrate industrial standards according to their purposes and features. Despite being published by official international organizations, different standards may contain divergent definitions for similar entities. Further, when utilizing the same standard for the design of a CPS, different views can generate interoperability conflicts. Albeit expressive, standardization frameworks may represent divergent categorizations of the same standard to some extent, interoperability conflicts need to be resolved to support effective and efficient communication in smart factories.

To achieve interoperability, data need to be semantically integrated and existing conflicts conciliated. This problem has been extensively studied in the literature. Obtained results can be applied to general integration problems. However, current approaches fail to consider specific interoperability conflicts that occur between entities in I40 scenarios. In this thesis, we tackle the problem of semantic data integration in I40 scenarios. A knowledge graph-based approach allowing for the integration of entities in I40 while considering their semantics is presented. To achieve this integration, there are challenges to be addressed on different conceptual levels. Firstly, defining mappings between standards and standardization frameworks; secondly, representing knowledge of entities in I40 scenarios described by standards; thirdly, integrating perspectives of CPS design while solving semantic heterogeneity issues; and finally, determining real industry applications for the presented approach.

We first devise a knowledge-driven approach allowing for the integration of standards and standardization frameworks into an Industry 4.0 knowledge graph (*I40KG*). The standards ontology is used for representing the main properties of standards and standardization frameworks, as well as relationships among them. The I40KG permits to integrate standards and standardization frameworks while solving specific semantic heterogeneity conflicts in the domain. Further, we semantically describe standards in knowledge graphs. To this end, standards of core importance for I40 scenarios are considered, i.e., the Reference Architectural Model for I40 (RAMI4.0), AutomationML, and the Supply Chain Operation Reference Model (SCOR). In addition, different perspectives of entities describing CPS are integrated into the knowledge graphs. To evaluate the proposed methods, we rely on empirical evaluations as well as on the development of concrete use cases. The attained results provide evidence that a knowledge graph approach enables the effective data integration of entities in I40 scenarios while solving semantic interoperability conflicts, thus empowering the communication in smart factories.

Contents

CHAPTER 1

Introduction

The currently ongoing digitization processes in many domains are generating data that increasingly influences many aspects of society. Globalization, the ubiquitous presence of communication networks and the Internet, new human-machine collaboration scenarios, e.g., social and professional networks, as well as the presence of complex information systems, are some of the activities that consume and generate large amounts of data. These activities are influencing practically all areas of society and industry. Data generated in these digitization processes can be of paramount importance for the improvement of many areas of human development. In particular, companies consider data increasingly as an asset of key relevance for enhancing the efficiency and efficacy of their processes.

In the engineering and manufacturing domain, there is currently an atmosphere of departure to a new era of digitized production. The *fourth industrial revolution* has been coined as "Industrie 4.0" in Germany, while related terms, e.g., "Industrial Internet" in USA, "Smart Manufacturing" in China, "Industrie du Future" in France, are used to denote the same concept in different countries. The term Industry 4.0 (I40) seems to be recognized by the international community to refer to the fourth industrial revolution. The main objective of I40 is the creation of *Smart Factories* by combining the advantages of the Internet of Things (IoT), Internet of Services (IoS) and Cyber-Physical Systems (CPS). In smart factories, humans, machines, materials as well as CPS need to cooperate in an intelligent manner to increase production.

To accomplish the objective of creating smart factories, three dimensions of data integration must be ensured: 1) vertical integration, within a factory/or production shop; 2) horizontal integration, through the entire value creation network; and 3) end-to-end integration across the entire product life-cycle [1, 2]. First, vertical integration comprises the integration of systems at various hierarchical manufacturing levels into one comprehensive solution. This integration is performed from the shop-floor level where devices such as sensors, actuators, CPS are located to the enterprise planning level with Enterprise Resource Systems (ERP). Second, horizontal integration, involves collaboration among partners, suppliers, customers but also other ecosystem members, from logistics to innovation, flows as well as stakeholders. Smart factories reach the globe by using worldwide production chains and data networks in their operations. Therefore, it is necessary to perform the integration of data between all these participants, i.e., horizontal integration. Horizontal integration needs to ensure that the factory is able to interact as a smart factory in a global marketplace. Finally, product life-cycle development comprises many engineering activities to create a CPS, e.g., conception, design, production, utilization, and termination. During the engineering of complex systems such as CPS, stakeholders typically belonging to different engineering disciplines, have to efficiently collaborate. The aim of the engineering process of a CPS is to deliver a high-quality end product, e.g., a complete production plant design, and to satisfy strict time frames. The presence of various engineering disciplines leads to a highly complex and software-intensive environments, which are characterized by a) a multitude of engineering tools that are not designed to cooperate with each other; b) a variety of engineering domain-specific representations and data exchange formats applied; and c) differences in the adopted workflows across the involved disciplines. Different systems, organizations, and stakeholders are involved in the engineering and operation of CPS both across engineering domain boundaries, i.e., horizontal integration, and between different abstraction levels (business, engineering, operation) of the system, i.e., vertical integration [3]. Furthermore, a core challenge in these environments is to ensure interoperability allowing for the integration of data throughout the entire product life-cycle. To achieve this interoperability, a key issue for realizing CPS relies in solving data integration challenges among these systems, organizations, and stakeholders. This means that to develop a CPS the affected disciplines must ensure the integration of the generated data.

Achieving such integrations is a complex task, in particular, when considering factories across all over the world, which typically operating according to different business and legal rules as well as different standards. Interoperability is a major challenge as well as one of the design principles of I40 [4]. To materialize interoperability in I40 scenarios, the meaning of entities like actuators, sensors, conveyors, and CPS, needs to be semantically described in a way that machines, as well as humans, are able to understand and share their meaning. When trying to cooperate in the described integrations, semantic interoperability conflicts among involved entities appear. Semantic interoperability conflicts denote differences in modeling of different or equivalent concepts and how these concepts are expressed [5]. The resolution of these interoperability conflicts across the different entities demands to be solved to make possible the I40 vision.

With the aim of solving the problem of interoperability in smart factories, standards and standardization frameworks have been proposed all over the world by industrial communities. Relevant examples in this regard are the Reference Architecture for Industry 4.0 (RAMI4.0) [6] or the Industrial Internet Reference Architecture (IIRA) [7]. Of fundamental value for these environments is to enable interoperability among CPS that are built based on these standardization frameworks. Albeit being expressive to categorize and align existing standards, standardization frameworks may present divergent interpretations or classifications of the same standard. For instance, OPC UA is classified by RAMI4.0 as a communication standard, whereas IIRA localizes OPC UA in the framework layer of its architecture [8]. Mismatches among standard classifications generate *conflicts* which negatively impact interoperability in smart factories. Thus, despite all these efforts for the creation of standards as well as standardization frameworks, semantic interoperability conflicts remain as an unsolved problem in I40 scenarios.

Interoperability among analogous I40 related standards is hampered due to different and/or similar representations of entities or processes. For instance, distinct names are used to express the same meaning in standards aiming to collaborate, e.g., an *InternalElement* in AutomationML (AML) describes the same meaning as an *Object* in OPC UA [9]. In case that those standards are jointly used to model the same CPS, the integration of their information models is required. To this end, the *meaning* of the entities involved demands to be precisely defined and the interoperability conflicts resolved [10]. In summary, to achieve the required interoperability in I40 scenarios, data need to be semantically integrated independently of the type of integration that is to be faced, i.e., vertical, horizontal or end-to-end engineering integration.

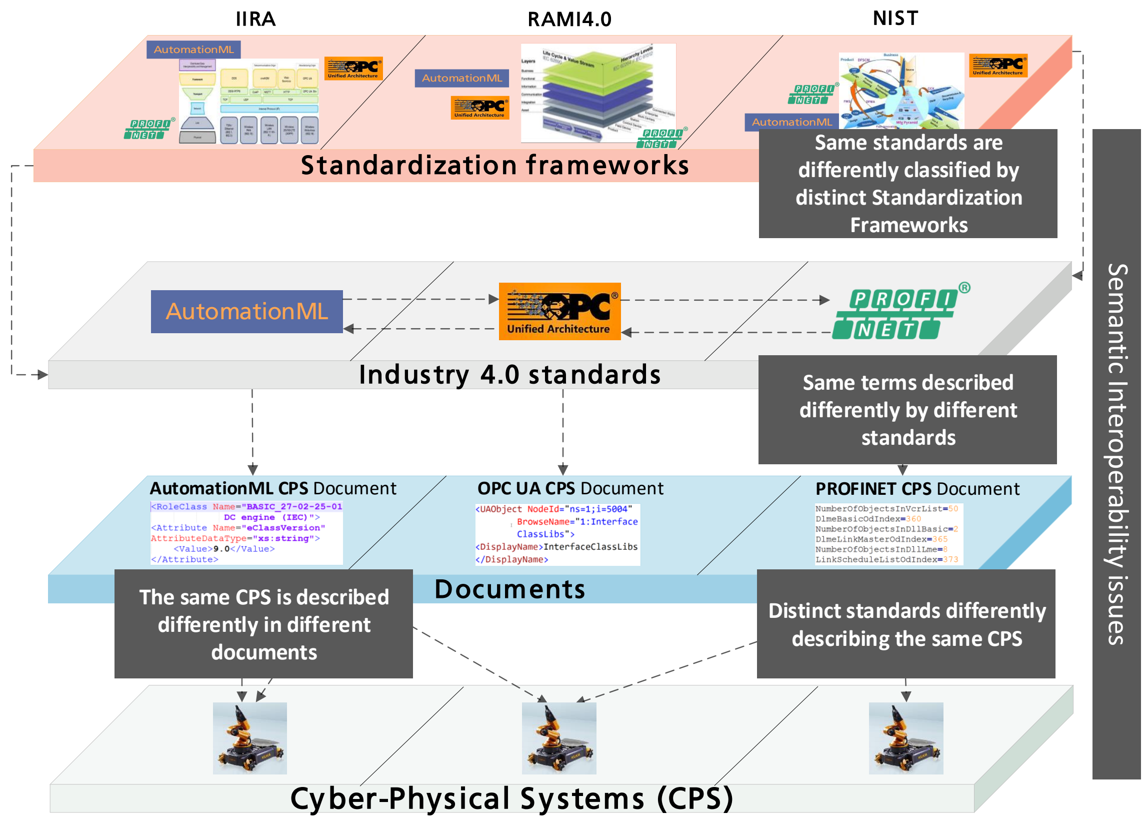


Figure 1.1: **Semantic interoperability conflicts in I40 scenarios**. Four levels have been defined to describe semantic interoperability conflicts in I40 scenarios. From Standardization Frameworks to the physical world represented by Cyber-Physical Systems. Semantic interoperability conflicts occur in all the levels negatively impacting data integration in I40 scenarios.

1.1 Problem Definition and Challenges

At the conceptual level, a semantic data integration problem is faced. The *research problem* guiding the work of this thesis can be expressed as follows: we investigate how interoperability in I40 scenarios can be enhanced by describing the meaning of entities in these scenarios.

Due to the reason that the problem of semantic data integration in I40 scenarios comprises many issues and obstacles to be addressed, we consider the following challenges and problems out of the scope of this thesis: big data challenges while semantically integrating data in I40 scenarios; security concerns of I40 scenarios; and real-time semantic data integration of I40 data. Despite of this, we acknowledge that the results presented in this thesis create the basis towards the extension of this work for covering also these aspects.

For a better comprehension of the semantic data integration problem in I40 scenarios, as well as the different semantic interoperability conflicts that need to be tackled, four levels have been identified (cf. Figure 1.1). The top level corresponds to *Standardization Frameworks*. In this level, standardization frameworks are investigated, such as RAMI4.0, IIRA, IICF, as well as the National Institute of Standards and Technologies (NIST) Standardization Landscape. The different categorization levels that standardization frameworks use for categorizing standards such as dimensions and layers are investigated. Further, we examined how standards are included in these levels with the goal to understand existing semantic interoperability conflicts that need to be addressed, e.g., same standards are differently classified by distinct standardization frameworks. The second level – *Industry 4.0 Standards*, takes relations between standards into account. This level also comprises semantic interoperability conflicts. For example, distinct names corresponding to the information models of two standards that are supposed to interact are used to express the same meaning of an entity, e.g., an *InternalElement* in AML has the same meaning as *Object* in OPC UA. The third level, *Documents*, refers to generated documents based on the standards describing features and relations of I40 entities, e.g., a CPS. These documents are built from different disciplines representing distinct views of the same CPS. Typically, semantic heterogeneity conflicts are introduced, i.e., various interpretations of the same domain are modeled. This is caused by the varying views involved in the process. The fourth level, *Cyber-Physical Systems*, describes the physical world, where CPS are in place. In the following, the main challenges tackled by this thesis are presented. The first three challenges refer to research while the fourth one focuses on applying the research to a concrete scenario.

Challenge 1: Defining mappings among standards and standardization frameworks. Standardization frameworks categorize standards according to their functions. However, standardization frameworks represent regional views regarding standards in I40 scenarios. Some standards, e.g., OPC UA, may be classified at different layers by RAMI4.0, IIRA, and the NIST standardization landscape. Therefore, different views about standards and standardization frameworks exist. Semantic conflicts between the representation of the standards with respect to different standardization frameworks need to be identified. Further, some standards are named differently by different standardization organizations, e.g., OPC UA is named IEC 62541 in its international version. These different representations of standards in standardization frameworks negatively impact the interoperability in I40 scenarios. Consequently, mappings among standardization frameworks and standards, as well as between standards are required to be identified.

Challenge 2: Representing knowledge about entities in Industry 4.0 scenarios. Standards comprise information models to represent the knowledge of the domain they cover. In some cases, these information models contain ambiguous, redundant, and overlapping information. Further, this information is encoded in semi-structured or unstructured formats, e.g., XML or plain text as well as structured formats, e.g., database models. Representing this knowledge in a computer-readable form that allows for the identification and solution of semantic interoperability conflicts among I40 entities is crucial for the work in this thesis.

Challenge 3: Integrating conflicting perspectives of entities in Industry 4.0 scenarios. CPS are complex systems that typically require input from several disciplines, such as mechanical, electrical or software engineering. Each one of these disciplines generates different views while designing a CPS. The different views need to be integrated into a final CPS design. In addition, entities individually modeled in each perspective, as well as the resolution of the corresponding semantic heterogeneity conflicts that may be caused, should be part of the final CPS design according to how consistent they are with respect to the other perspectives.

Challenge 4: Determining real-world applications for semantic data integration of entities in Industry 4.0. Interoperability and semantic data integration are recognized design principles and requirements for the development of the I40 vision. However, determining real-world applications where the added value of semantic-based approaches can be demonstrated is difficult due to: 1) lack of understanding of semantic heterogeneity conflicts of the data generated in the I40 contexts; 2) usage of standards that are not sufficiently expressive enough to solve the problem of semantically integrating data in I40 contexts, e.g., XML; and 3) the absence of success stories demonstrating the benefits of semantic-based approaches for data integration.

1.2 Research Questions

Following the discussion in the previous sections, the following research questions are defined.

> **RQ1**: How can a knowledge graph approach define mappings of standards and standardization frameworks and resolve existing semantic interoperability conflicts among them?

In order to answer this research question, a knowledge graph approach is used to represent and integrate knowledge encoded in various standardization frameworks and standards. With this approach, semantic interoperability conflicts among standardization frameworks and standards are conciliated.

> **RQ2**: How can knowledge graphs represent semantics encoded in Industry 4.0 entities?

To respond to this question, ontologies covering different areas of the I40 domain are developed. The benefits provided for this approach, compared to traditional knowledge management approaches are demonstrated.

> **RQ3**: How can existing rule-based approaches be utilized to resolve semantic interoperability conflicts over knowledge graphs?

To answer this research question we investigated logic programming approaches and probabilistic techniques for creating and exploiting knowledge graphs. The logic programming approaches and probabilistic techniques are employed for capturing the knowledge encoded in different CPS perspectives. This knowledge is encoded in knowledge graphs and exploited with the objective of identifying semantic interoperability conflicts between CPS perspectives. Then, semantic interoperability conflicts are solved by relying on the knowledge graphs. Finally, the final design representing the integrated knowledge of the CPS perspectives is created.

> **RQ4**: How can a knowledge graph-based integration of entities be applied in Industry 4.0 real-world scenarios?

For addressing this question, different application areas for semantic data integration in I40 scenarios are examined. Real use cases have been developed and practical experiences in a manufacturing company using this application are reported.

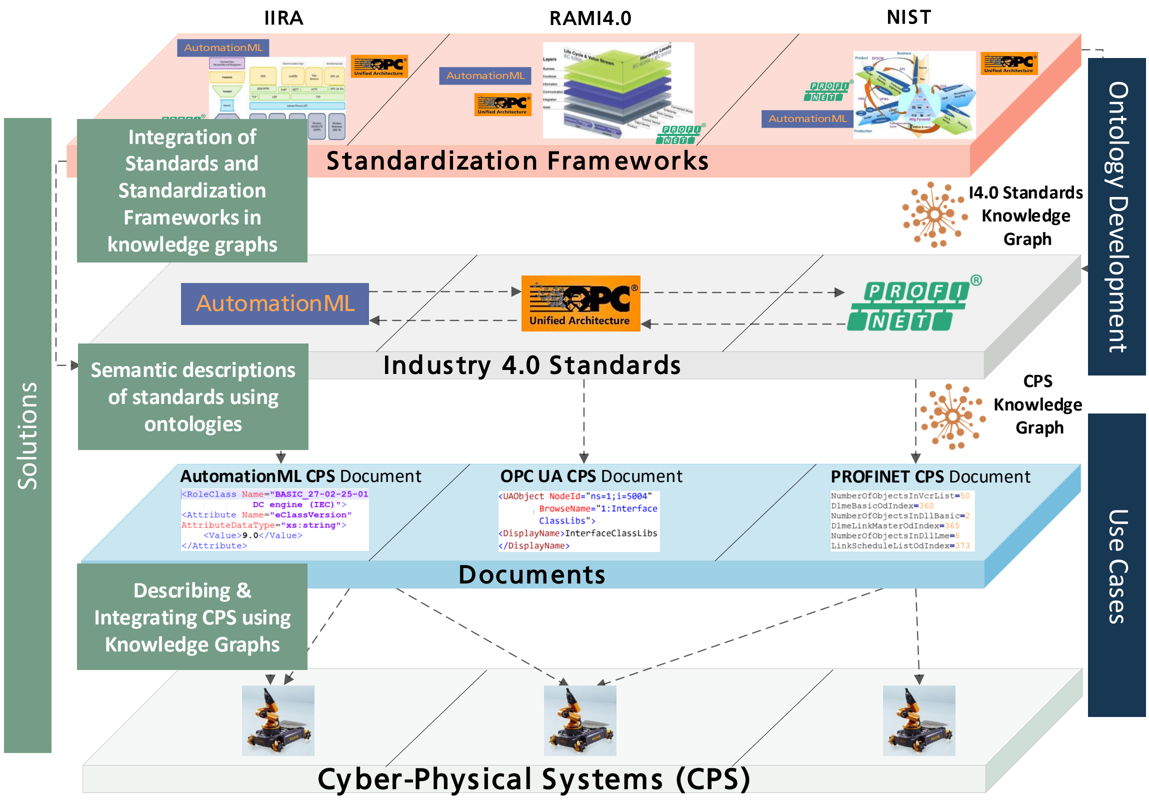


Figure 1.2: **Thesis Contributions.** The four contributions of this thesis propose solutions for reconciling interoperability conflicts in Industry 4.0 scenarios based on knowledge graphs. 1) Integrating standards and standardization frameworks into a knowledge graph; 2) Semantically describing standards using ontologies; 3) Integrating CPS into knowledge graphs; and 4) A practical application of the proposed knowledge graph-based approach for semantically integrating data in I40 scenarios. The semantics of standards and standardization frameworks are encoded in knowledge graphs to solve semantic interoperability conflicts in Industry 4.0 scenarios.

1.3 Thesis Overview

In order to guide the reader throughout this document, we present an overview of the main contributions and the research areas covered in this thesis. Additionally, references to scientific publications supporting this work are included.

1.3.1 Contributions

The contributions of this thesis are cross disciplinary involving semantic modeling, knowledge graph creation and refinement, as well as semantic data integration in I40 scenarios. Figure 1.2 describes the proposed solutions for integrating data while solving semantic interoperability conflicts in I40 scenarios according to the identified levels. Next, the contributions of this thesis are outlined.

Knowledge graphs (KGs) have proven to be successful to cope with semantic interoperability conflicts during data integration in different domains such as medicine [11], agriculture [12], and human traffic [13]. For factories, KGs are considered to be at the core of the next generation of Enterprise Information Systems [14]. The meaning of data is stored alongside the graph, in the form of ontologies capturing the semantics of the domain. KGs also enable drawing conclusions and new knowledge based on the existing one. This makes KGs a single place to find and to understand data. In order to achieve semantic interoperability, data described by standards and standardization frameworks require to be semantically integrated. The meaning of these data need to be preserved and semantic heterogeneity conflicts are required to be addressed during the integration.

- **Contribution 1:** *Integrating standards and standardization frameworks into a knowledge graph.* A knowledge-graph based approach to semantically integrate documents adhering to I40 standards and standardization frameworks is proposed. The *STO* ontology, that describes the concept of standards and standardization frameworks to is developed. Further, a methodology to build and exploit a knowledge graph of Industry 4.0 standards and standardization frameworks is presented. Based on this methodology, and the semantics represented in *STO*, we build the Industry 4.0 standards knowledge graph (I40KG). The I40KG is populated with descriptions of more than 200 standards, more than 25 standardization organizations, and 100 relations between the standards. Finally, the I40KG has been linked to existing knowledge graphs such as DBpedia and an automated reasoning has been implemented to reveal implicit relations between standards as well as mappings across standardization frameworks. This contribution aims to answer **RQ1**.
- **Contribution 2:** *Semantically describing standards using ontologies.* For the second level, i.e., the *Industry 4.0 standards*, the semantic description of standards using ontologies is proposed. A novel approach to semantically represent and exploit knowledge of standards and standardization frameworks related to I40 is outlined. Standards of paramount importance for the I40 vision are modeled as ontologies. First, RAMI4.0 covering the reference architecture for I40 solutions and the Administration Shell concept which provides a representation of assets. Second, the AML ontology, which covers the AutomationML standard. This standard is crucial in industry solutions for designing CPS from distinct discipline perspectives such as the mechanical, electrical and software engineering ones. Finally, SCORVoc representing the supply chain operations reference model of the APICS industry association. We demonstrate the benefits of the semantic representation of Industry 4.0 entities. Common use cases of the semantic representation in I40 scenarios are developed, e.g., the units of measurements. The codification of semantic heterogeneity conflicts among entities in these scenarios is introduced. Furthermore, the solution of conflicts by considering and applying the semantics of the ontologies is developed. This contribution covers research question **RQ2**.
- **Contribution 3:** *Integrating CPS into knowledge graphs.* We propose an approach for integrating CPS perspectives into knowledge graphs. The knowledge graphs are created for representing the information from different perspectives of CPS design, i.e., mechanical, electrical, and software views. The semantic interoperability conflicts that occur between the perspectives are characterized. For this purpose, we formalize the problem of identifying and solving conflicts among I40 entities of CPS perspectives following two logical approaches: the Deductive Databases and the Probabilistic Soft Logic. The specifications of these formalizations are implemented in Alligator and SemCPS, respectively. First, we presented Alligator, a deductive approach for the identification

and solution of semantic interoperability conflicts between CPS documents. ALLIGATOR relies on Datalog to *accurately* represent the knowledge that characterizes different types of semantic heterogeneity conflicts in CPS documents. ALLIGATOR uses a knowledge graph to encode the knowledge of the CPS perspectives. Second, we developed SEMCPS, a rule-base framework that relies on Probabilistic Soft Logic (PSL) for capturing the knowledge encoded in different CPS perspectives and exploiting this knowledge for CPS perspective integration while solving existing semantic heterogeneity conflicts. Regarding the *Document* level, as well as for the *Cyber-Physical Systems* level, we aim to create a CPS KG able to describe and integrate CPS documents defined by different standards. With this proposal, research question **RQ3** is addressed.

- **Contribution 4:** To showcase the applicability of the knowledge graph approach as a contribution, a case study based on a manufacturing company is performed. Two use cases of core importance for the efficiency of factory production are developed, i.e., tool availability and energy consumption. We investigated the data sources of the manufacturing company that are related to the use cases. Existing semantic interoperability conflicts among the data sources are analyzed. To execute the use cases, we developed a knowledge graph approach for the solution of the semantic interoperability conflicts existing between the data sources of the company. A set of ontologies was developed to describe the semantics of the data sources, i.e., bill of material, manufacturing execution systems and sensor data. In addition, a set of mappings are defined to map the data sources with the ontologies. An architecture for implementing the knowledge graph approach is defined. The architecture enables the integration of data considering the data sources, ontologies, mappings and applications. By using the proposed approach semantic interoperability conflicts between the data sources are resolved. **RQ4** is answered with the results obtained in this contribution.

1.3.2 Publications

Parts of the work presented in this thesis have already been published as conference, workshop and journal articles or book chapters. At the beginning of each chapter, the publications which the chapter is based on are referenced. In the following, the main publications building the basis of this thesis are outlined.

1. **Irlán Grangel-González**, Lavdim Halilaj, Gökhan Coskun, Sören Auer. *Towards Vocabulary Development by Convention.* In Proceedings of the International Conference on Knowledge Engineering and Ontology Development (KEOD), 2015, 334-343, SciTePress; This article is a joint work with Lavdim Halilaj, a PhD student at the University of Bonn. In this article, I contributed to the definition of the problem, the development of the vocabulary development method, the evaluation, as well as the analysis of the results.

2. **Irlán Grangel-González**, Lavdim Halilaj, Gökhan Coskun, Sören Auer, Diego Collarana, Michael Hoffmeister. *Towards a Semantic Administrative Shell for Industry 4.0 Components.* In Proceedings of the Tenth IEEE International Conference on Semantic Computing (ICSC) 2016, 230-237, IEEE. Fraunhofer IAIS Paper of the Month, June 2016. This article is a joint work with Lavdim Halilaj, a PhD student at the University of Bonn. In this article, I contributed to the definition of the problem, the development of the

approach, the review of state-of-the-art approaches, the presentation of the use cases, as well as the analysis of the results.

3. Lavdim Halilaj, **Irlán Grangel-González**, Gökhan Coskun, Sören Auer. *Git4Voc: Git-based Versioning for Collaborative Vocabulary Development.* In Proceedings of the Tenth IEEE International Conference on Semantic Computing 2016, 285-292, IEEE; This article is a joint work with Lavdim Halilaj, a PhD student at the University of Bonn. In this article, I contributed to the definition of the problem, the development of the method, the critical review to the state-of-the-art, as well as the analysis of the results.

4. **Irlán Grangel-González**, Lavdim Halilaj, Gökhan Coskun, Sören Auer, Diego Collarana. *An RDF-based approach for implementing Industry 4.0 components with Administration Shells.* In Proceedings of the 21st IEEE International Conference on Emerging Technologies and Factory Automation (EFTA) 2016, 1-8, IEEE. This article is a joint work with Lavdim Halilaj, a PhD student at the University of Bonn. In this article, I contributed to the definition of the problem, the development of the approach, the review of state-of-the-art approaches, the presentation of the use cases, as well as the analysis of the results.

5. Niklas Petersen, **Irlán Grangel-González**, Sören Auer, Gökhan Coskun, Marvin Frommhold, Sebastian Tramp, Maxime Lefranc, Antoine Zimmermann. *SCORVoc: Vocabulary-based Information Integration and Exchange in Supply Networks.* In Proceedings of the Tenth IEEE International Conference on Semantic Computing 2016, 132-139, IEEE; This article is a joint work with Niklas Petersen, a PhD student at the University of Bonn. My contributions to this article are dedicated to the problem definition, ontology modeling, as well as analysis and review of related work.

6. **Irlán Grangel-González**, Diego Collarana Vargas, Lavdim Halilaj, Steffen Lohmann, Christoph Lange, Maria-Esther Vidal, Sören Auer. *Alligator: A Deductive Approach for the Integration of Industry 4.0 Standards.* In Proceedings of the 20th International Conference of Knowledge Engineering and Knowledge Management (EKAW) 2016, 272-287; This article is a joint work with Diego Collarana Vargas and Lavdim Halilaj, both PhD students at the University of Bonn. In this article, I contributed to the definition of the problem and motivating example, the development of the approach, the revision of the state-of-the-art approaches, the development of the software, as well as the execution and analysis of the experiments and results.

7. **Irlán Grangel-González**, Paul Baptista, Lavdim Halilaj, Steffen Lohmann, Maria-Esther Vidal, Christian Mader, Sören Auer. *The Industry 4.0 Standards Landscape from a Semantic Integration Perspective.* In Proceedings of the 21st IEEE International Conference on Emerging Technologies and Factory Automation 2017, 1-8; In this article, my contributions are the definition of the problem and motivating example, the development of the approach, the development of the ontology and the knowledge graph, the revision of the state-of-the-art approaches, as well as the execution and analysis of the experiments and results.

8. **Irlán Grangel-González**, Lavdim Halilaj, Omar Rana, Maria-Esther Vidal, Steffen Lohmann, Sören Auer, Andreas W. Müller. *Knowledge Graphs for Semantically Integrating of Cyber-Physical Systems.* In Proceedings of the 29th International Conference of Database and Expert Systems Applications (DEXA) 2018, 184-199. In this article, I contributed to

the definition of the problem and motivating example, the development of the approach, the software implementation, the review of related work, as well as the execution and analysis of the experiments and results.

9. Niklas Petersen, Lavdim Halilaj, **Irlán Grangel-González**, Steffen Lohmann, Christoph Lange, Sören Auer. *Realizing an RDF-based Information Model for a Manufacturing Company – A Case Study.* (One of the two nominees for the Best In-Use Paper Award) In Proceedings of the 16th International Semantic Web Conference (ISWC) 2017, 350-366, Springer. This is a joint work with Niklas Petersen and Lavdim Halilaj both PhD students at the University of Bonn. In this article, I contributed to the development of the information model, the definition of the mappings, the development of the architecture, the description of the use cases, as well as the analysis of how semantic heterogeneity conflicts can be solved with the information model.

A complete list of publications completed during the PhD term is available in Appendix A.1.

1.4 Thesis Structure

The thesis is structured in seven chapters, outlined as follows.

- *Chapter 1 - Introduction* prefaces the thesis covering the main research problem and challenges, motivation for the conducted work, research questions, scientific contributions that address research questions, and a list of published scientific articles that formally describe those contributions.
- *Chapter 2 - Background and Preliminaries* introduces the key concepts required to understand the work of this thesis. Initially, I40 scenarios and core related concepts are explained. Next, the foundations of semantic technologies are described. General principles of data integration are examined with focus on semantic data integration; semantic heterogeneity conflicts and their presence in I40 scenarios are presented. Finally, the description of techniques used to integrate data while solving semantic heterogeneity conflicts is investigated.
- *Chapter 3 - Related Work* examines current state-of-the-art approaches to provide the reader a better comprehension of the work conducted in this thesis. General approaches for semantic data integration are investigated. Further, specific works with respect to the semantic representation of standards for the I40 domain are described. Next, works regarding the integration of standards into knowledge graphs are outlined. Finally, existing methods for the semantic integration of entities in the I40 domain are described.
- *Chapter 4 - Integrating Industry 4.0 Standards into a Knowledge Graph* describes a knowledge graph approach to solve interoperability conflicts among standardization frameworks as well as among standards; and also outlines a methodology to build and refine knowledge graphs.
- *Chapter 5 - Semantically Describing Industry 4.0 Standards using Ontologies* presents a methodology for building ontologies of entities described with standards that are commonly utilized in I40 scenarios, i.e., RAMI4.0, AML, and SCOR. This methodology is employed to leverage the semantics of entities encoded in these standards and helps to solve semantic heterogeneity issues.

- *Chapter 6 - Integrating Cyber-Physical Systems into Knowledge Graphs* outlines the integration of CPS into knowledge graphs. Two solutions are presented to the problem tackled in this chapter: i) a deductive approach combining the power of Datalog and ontologies; and ii) an approach considering the uncertainty present in CPS design and using probabilistic soft logic methods to obtain the most probable design of a CPS.

- *Chapter 7 - Applications of Semantic Data Integration to Industry 4.0 Scenarios* shows the applicability of the knowledge graph approach for semantically integrating data in an actual manufacturing company.

- *Chapter 8 - Conclusion and Future Direction* finalizes this thesis with a summary of the results and contributions to the problem of semantic interoperability in I40 scenarios. Existing limitations of the presented approach are discussed and an outlook on possible directions for future research is provided.

CHAPTER 2

Background and Preliminaries

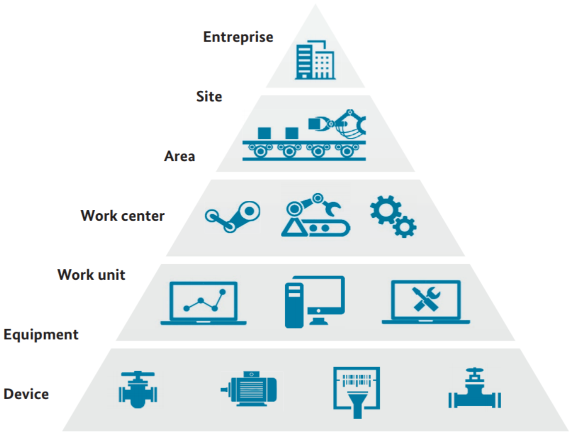
This chapter outlines the background of the work conducted in this thesis. Section 2.1 presents Industry 4.0 and related concepts such as Cyber-Physical Systems, standards, and standardization frameworks. Semantic Technologies utilized to represent ontologies and knowledge graphs, e.g., RDF, OWL, SPARQL are described in Section 2.2. Moreover, the semantic data integration concept is investigated. Semantic heterogeneity conflicts are exposed with a particular interest in those impacting on Industry 4.0 scenarios. Finally, existing approaches for integrating data while solving semantic heterogeneity conflicts are examined in Section 2.3.

2.1 Industry 4.0

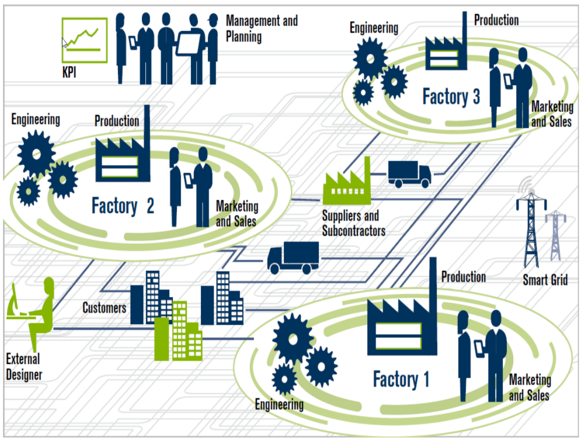
Industry 4.0 (I40) is the information-intensive transformation of manufacturing and other industries in a connected environment of data, people, processes, services, systems, and IoT-enabled industrial assets. I40 utilizes actionable information as a way and means to realize smart industry and ecosystems of industrial innovation and collaboration.[1] The main objective of I40 is to drive manufacturing to be more efficient by optimizing and personalizing production processes. I40 is based on the concept of smart factories, where the machines are integrated with humans through CPS [15]. Smart factories are able to automatically exchange information between manufacturing resources such as sensors, actuators, machines, robots, and conveyors. They can also intelligently maintain production process and be self-sustainable. To accomplish the objective of creating smart factories, three types of integration must be ensured: 1) Vertical integration, within a factory/or production shop; 2) Horizontal integration, through value networks; and 3) End-to-End engineering integration across the entire value chain to support product customization [2].

Vertical Integration in I40 Vertical integration is related to the integration of IT systems at various hierarchical production and manufacturing levels into one comprehensive solution. This integration is performed within a factory, and is typically described by the automation pyramid (cf. Figure 2.1(a)). The automation pyramid comprises the levels considered for vertical integration, i.e., from the physical devices to the enterprise.

[1] https://www.i-scoop.eu/industry-4-0/



(a) Automation Pyramid



(b) Horizontal value chain

Figure 2.1: **Horizontal and vertical settings of I40 scenarios.** Automation pyramid showing layers required to be integrated, i.e., from the Product to the Enterprise, taken from [2]. Horizontal value chain interaction among diverse areas of factories, suppliers, and customers, taken from [17].

Horizontal Integration in I40 Horizontal integration refers to the integration of IT systems for and across the various production and business planning processes (cf. Figure 2.1(b)). Horizontal integration is about digitization across the full supply chain. This involves seamless integration and data exchange with suppliers, customers, and external stakeholders.

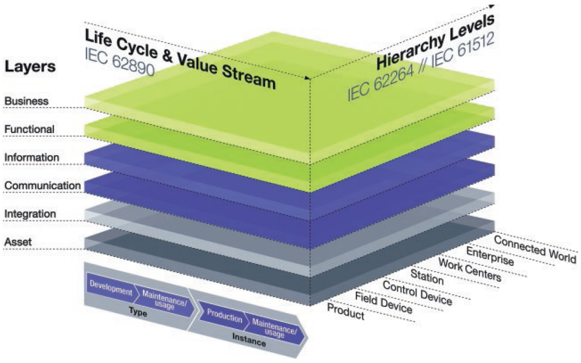
End-to-End Integration With the aim to deliver high-quality end products and to satisfy tight time-frames, a chain of activities is involved. These activities include customer requirement expression, product design and development, production planning, production engineering, production, services, maintenance, and recycle. Several disciplines are involved in the development of these activities generating data that need to be integrated. By integration, a continuous and consistent product model can be reused by every stage [16]. I40 builds upon data models and data mapping across the mentioned end-to-end product life-cycle and value stream. All technologies in I40 need to be seen in that perspective, whereby integration is key.

2.1.1 Cyber-Physical Systems

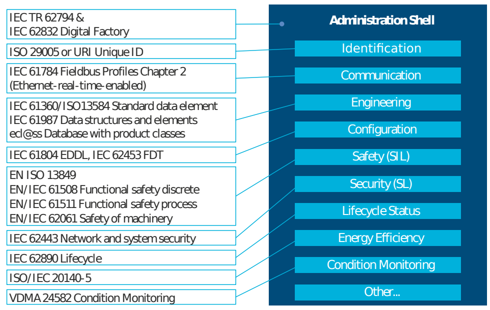
Cyber-Physical Systems (CPS) are at the core of the I40 movement. CPS are defined as a set of heterogeneous physical units, e.g., sensors, control modules, communicating via networks and interacting with applications deployed on cloud infrastructures and/or humans to achieve a common goal [18]. CPS integrate an IT system with mechanical and electronic components connected to online networks that allow the communication between machines in a similar way to social networks [19]. Typically, CPS comprise three main parts, i.e., a cyber part as a computing core, a physical part as a controlled object, and a network for establishing the communication between the cyber and the physical part [5].

2.1.2 Standardization Frameworks for Industry 4.0

Several standardization frameworks have been developed worldwide to provide a unified view for I40. All these standardization frameworks pursue the common objective of providing a



(a) RAMI4.0



(b) Administration Shell

Figure 2.2: **The RAMI4.0 Administration Shell concept (adjusted from [20]).** (a) I40 Assets are enclosed into the RAMI4.0 Administration Shell, e.g., a motor. (b) Alignments between I40 standards and the RAMI4.0 Administration Shell concept and I40 submodels (taken from [6]). The Administration Shell provides another classification for standards with focus on I40 assets. This classification also generates semantic interoperability conflicts among different specifications of the Administration Shell and standards used in the submodels, e.g., standards to be used in the identification submodel, i.e., ISO 29005 [6] and ISO 11179 [2].

roadmap for the use of standards in the context of smart factories. This section examines the most relevant standardization frameworks for the development of this work.

Reference Architecture Model for Industry 4.0

The Reference Architecture Model for Industry 4.0 (RAMI4.0) encompasses the core aspects of Industry 4.0 in a three-dimensional layer model [6, 20]. It illustrates the connection between IT, manufacturers/plants and the product life-cycle in a three-dimensional space. Each dimension shows a particular part of these domains divided into different layers, as depicted in Figure 2.2(a). The model extends the hierarchy levels defined in *IEC 62264/61512* by adding the concepts *Product* on the lowest level and *Connected World* at the top level, which goes beyond the boundaries of an individual factory.

The vertical axis on the left-hand side of Figure 2.2(a) represents the IT perspective, comprising layers ranging from the physical device (*asset*) to complex functions as they are available in ERP systems (*functional*). These layers correspond to the IT way of thinking, where complex projects are decomposed into smaller manageable parts. The horizontal axis on the left-hand side indicates the product life-cycle where *Type* and *Instance* are distinguished as two main concepts. RAMI4.0 enables the representation of data gathered during the entire life-cycle. The horizontal axis on the right-hand side organizes the locations of the functionalities and responsibilities in a hierarchy. The concept of the *Administration Shell* is of core relevance in RAMI4.0. The Administration Shell plays a pivotal role in reaching the desired interoperability of a given asset. As asset is defined as a physical or logical object which is managed by an organization and which has a value for the organization [20]. In [20], the term asset is used to refer to an individual physical or non-physical entity. An asset can be an entire machine, an automation component, or a software platform; it can be a legacy system or a new system. The Administration Shell is capable of representing all the information of an asset during its complete life-cycle (cf. Figure 2.2(b)). As such the Administration Shell is responsible for exposing and conveying only those information structures, methods for interaction with, and capabilities of an asset that are both required for asset employment and permitted by its manufacturer. Hence, it essentially depicts a *smart interface* to an asset, which may individually vary in its extent but always provides a standardized access point for knowledge discovery and interaction orchestration.

Industry 4.0 Component

A component is a core concept in the Industry 4.0 context. As defined in [20], an I40 component constitutes a specific case of a CPS. It is used as a model to represent the properties of a CPS, e.g., real assets in a production environment connected with virtual assets and processes. An I40 component can be a production system, an individual machine, or an assembly inside a machine. It comprises two foundational elements, i.e., an asset and its Administration Shell. Every asset that has an associated Administration Shell becomes an I40 component.

Industrial Internet Reference Architecture

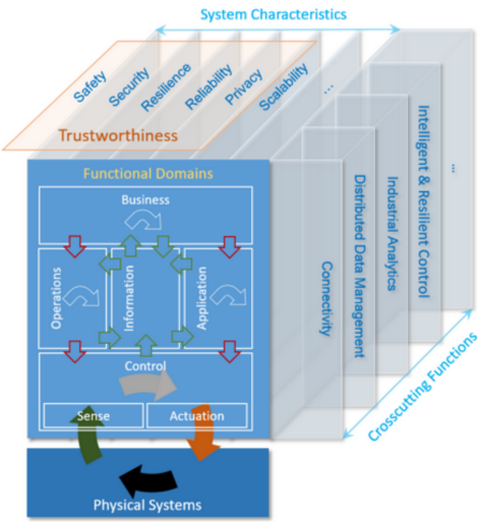
IIRA is a standards-based open architecture for Industrial Internet of Things (IoT)-based systems [7]. IIRA presents a generic description and representation with a high level of abstraction to support smart industry. It provides a framework comprising methods to design industrial internet systems, without making specific recommendations for standards that comprise these systems [21]. IIRA comprises the industrial internet viewpoints, i.e., business, usage, functional and implementation. The aim of these viewpoints is to provide an analysis of individual sets of IoT-based systems. Further, the Industrial Internet Connectivity Framework (IIRC) extends IIRA to map existing standards with different functional levels [8]. These levels range from the physical, link, network, transport, framework and the top level of distributed data interoperability and management.

National Institute of Standards and Technology

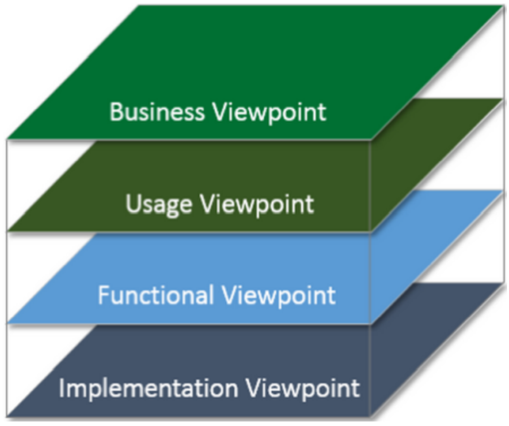
The NIST has defined a standards landscape with a focus on Smart Manufacturing Systems [22]. Two major classifications have been done in this work. First, the classification of standards regarding three manufacturing-related life-cycles: 1) product development life-cycle standards; 2) production system life-cycle standards; and 3) business cycle for supply chain management. Second, the classification regarding the ISA95 manufacturing pyramid, which classifies standards into five levels, i.e., from the device to the enterprise level. In the following, we detail the classification of standards given by NIST.

Product development life-cycle standards In this criteria of classification, standards are organized in different phases of the product life-cycle, such as Modeling Practice, Product Model and Data Exchange, Manufacturing Model Data, Product Category Data, and Product Life-cycle Data Management.

Production system life-cycle standard In this case, the classification of standards includes categories such as Production System Model Data and Practice, Production System Engineering, Production life-cycle data management, and Production System Operation and Maintenance.



(a) IIRA Functional Domains



(b) IIRA Viewpoints

Figure 2.3: **IIRA functional domains and viewpoints** (taken from [7]). Figure 2.3(a) depicts the IIRA relationships between the functional domains, crosscutting functions, and key system characteristics. Figure 2.3(b) shows the four IIRA viewpoints. This establishes the basis for a detailed viewpoint-by-viewpoint analysis of individual sets of IoT system concerns.

Business cycle for supply management This classification considers the cycle of supply chains, i.e., Plan-Source-Make-Deliver-Return. In general, standards utilized for modeling and executing business processes are included.

2.1.3 Industry 4.0 Related Standards

There exist a huge variety of standards related to I40. Standards are typically formal documents describing specific areas and are created by exiting standardization organizations. In I40 scenarios, standards enable the description of the properties of industrial components, systems, and processes, as well as interactions among them. In the following, we describe some of the most significant standards for the development of this thesis.

AutomationML

The *AutomationML* standard (AML) [23] enables modeling systems from single automation components to entire large and complex production systems and supports the representation of the various aspects of such systems, i.e., system's topology, geometry, kinematics, and control behavior [24, 25]. AML is an open (specification and schema are available), neutral (manufacturer independent without proprietary interfaces or libraries) and XML-based data exchange format that aims to ensure consistent and lossless data exchange during manufacturing systems design. AML is currently well recognized by major manufacturing companies such as Daimler, Audi, and Siemens and continues gaining acceptance from the manufacturing market players. Yet, as an XML-based standard lacks a formal semantic basis that is increasingly necessary for industrial projects [26, 27]. AML is not a completely new format, but rather consists of existing formats, which were extended, adapted, and combined appropriately. Such approach allows modeling manufacturing system data sequentially, i.e., starting from the plant structure design, and then adding the geometry and kinematics information up to process sequences and logical dependencies following the sequence of engineering disciplines involved in the engineering chain. The top level of AML is represented in terms of *Computer Aided Engineering Exchange* (CAEX, IEC 62424) format for plant topology, which is used for storing hierarchical object information, properties, and libraries [28]. The geometry (mechanical drawings) and kinematics (physical properties such as force, speed, or torsion) are implemented with the *COLLAborative Design Activity* (COLLADA) format [29]. Further, the logic, i.e., sequencing, behavior, and control information is implemented with *PLCopen* XML (IEC 61131).

OPC Unified Architecture

OPC Unified Architecture (OPC UA) is the next generation technology for OPC foundation which provides reliable and secure transportation of information to every authorized application and person at any time and in any place. The architecture consists of an asynchronous protocol, which is based on TCP, HTTP, or SOAP that are used to exchange messages over a network session. It has an XML based encoding scheme and provides secure communication channels. Furthermore, the architecture supports object orientation and semantic relations for the modeling of the information.

Supply Chain Operation Reference Model

The Supply Chain Operation Reference Model (SCOR) is an international standard to represent the processes and entities along Supply Chains (SC). The motivation behind SCOR is to enable enterprises to diagnose and manage their SC. It is challenging to agree on a standardized way to represent knowledge about the business processes and the supply network. This is partly due to the variety in company size, industry and business models, viewpoints, and granularity of requirements. The *APICS Supply Chain Council*[2] faced this challenge and elaborated the SCOR reference model [30].[3]

The main concept in SCOR is named *process*, and denotes any activity related to production and logistics. The SCOR model has different conceptualization levels. The Top Level contains the main processes: *Enable*, *Make*, *Source*, *Deliver*, *Return*. Then, the Configuration Level provides a set of process categories for main processes. Finally, the Process Element Level decomposes the process categories by adding process element definitions and process element information. This leads to a total of 201 definitions of industry-agnostic processes.

The focus of our work was on the SCOR model in its 11t revision [31–34]. SCOR has become a mature reference model backed up by many global players (including IBM, HP, and SAP). It contains industry-agnostic definitions for 201 processes and 286 metrics. Figure 2.4 depicts a high-level overview of the reference model. For that purpose, SCOR defines different performance indicators (metrics) including a calculation plan to ensure comparability within the entire Supply Chain. In total, there are 286 metrics which are grouped into five categories: *Reliability*, *Responsiveness*, *Agility*, *Costs* and *Assets*. The usage of these metrics allows Supply Chain managers to identify weak and strong links within the Supply Chain.

[2] http://www.apics.org/sites/apics-supply-chain-council/about-apics-scc
[3] http://www.apics.org/sites/apics-supply-chain-council/frameworks/scor

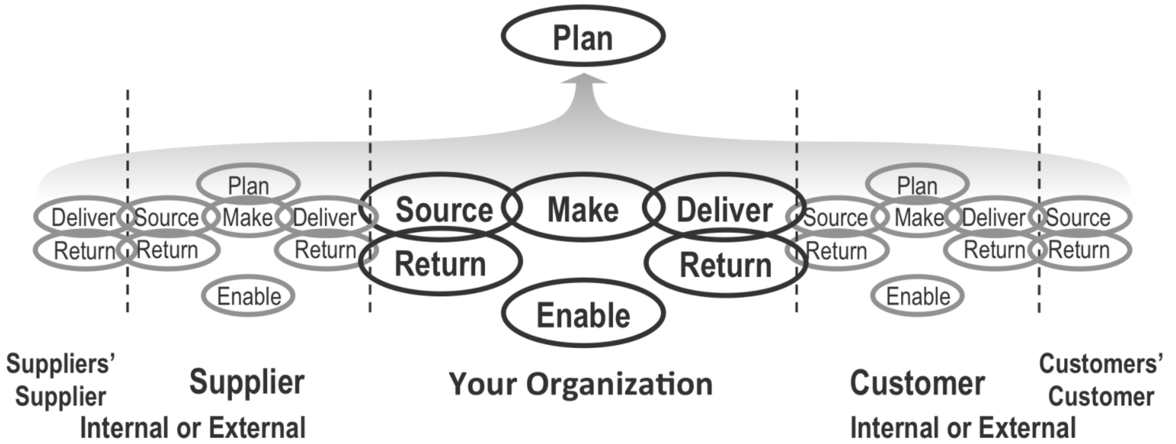


Figure 2.4: **High-level overview of the Supply Chain Organizations Reference in its version 11.0** (taken from [30]). The figure depicts the process view of the SCOR model. It shows the different stakeholders involved in the Supply Chain, i.e., Supplier's Supplier, Supplier, Customer and Customer's Customer. The organization under study is located at the center.

2.2 Semantic Technologies

The vision of the semantic technologies is to extend the World Wide Web by bringing structure to the meaningful content, so that it allows computers and people to better work in cooperation. In this section, main semantic technologies are described.

2.2.1 The Resource Description Framework

The Resource Description Framework (RDF) is a generic data model for interchanging data on the Web recommended by the World Wide Web Consortium (W3C)[4]. In RDF, data is represented as triples consisting of subjects, predicates, and objects, which can be combined to directed graphs composed of vertices representing subjects and objects and edges representing predicates. Formally, an RDF triple is defined as follows:

Definition 2.1: RDF Triple [35]

Let I, B, L be disjoint infinite sets of URIs, blank nodes, and literals, respectively. A tuple $(s, p, o) \in (I \cup B) \times I \times (I \cup B \cup L)$ is denominated an RDF triple, where s is called the subject, p the predicate, and o the object.

An example RDF graph representing information about a conveyor belt is shown in Figure 2.5. The resource `oi40:Conveyor_Belt` is of type machine. This is represented by the `rdf:type` property which connects, in this case, two resources, i.e., the `oi40:Conveyor_Belt` acting as a subject and the `oi40:Machine` acting as an object. Similarly, the resource `oi40:Thomas_Robins` is declared of type `oi40:Person` through the `rdf:type` property. In addition, this RDF graph represents that the Conveyor Belt was invented by Thomas Robins. International Resource Identifiers (IRIs) are used to identify resources unambiguously, while *literals* consisting of either

[4] https://www.w3.org/RDF/

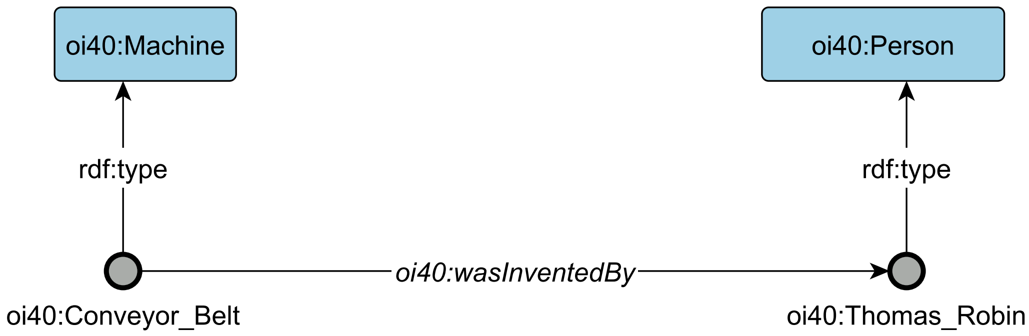


Figure 2.5: **Example of an RDF graph representing information about a Conveyor Belt.** The resource Conveyor Belt is a machine. The Conveyor Belt is connected through the property `was invented by` to the resource Thomas Robin which is defined as a person.

a string and language tag or a value and datatype describe concrete data values. To describe the examples, the notation `prefix:element` is used; prefix refers to the identification of the IRI and element can refer to the name of one of the elements of RDF, i.e., a subject, predicate or object. Formally, an RDF graph D is defined as a set of triples: $D \subset I \times I \times (I \cup L)$, where I represents the set of IRIs and L the set of literals. RDF can be serialized in different formats, such as RDF/XML[5], Turtle[6], RDFa[7] or JSON-LD[8]. Every serialization has their own pros and cons, depending on the use case. Throughout this document the Turtle notation is used because it favors the readability of RDF documents.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix oi40: <https://w3id.org/i40/ont/> .

 oi40:Conveyor_belt  rdf:type  oi40:Machine .
 oi40:Conveyor_belt  oi40:wasInventedBy  oi40:Thomas_Robins .
 oi40:Thomas_Robins  rdf:type  oi40:Person .
```

Listing 2.1: **Turtle serialization of the RDF graph in Figure 2.5**

2.2.2 Ontologies, RDF Schema, and the Web Ontology Language

In this work, an ontology is defined as a formal, explicit specification of a shared conceptualization [36]. This definition is analyzed and extended as follows:

5 https://www.w3.org/TR/rdf-syntax-grammar/
6 https://www.w3.org/TR/turtle/
7 https://www.w3.org/TR/rdfa-syntax/
8 https://www.w3.org/TR/json-ld/

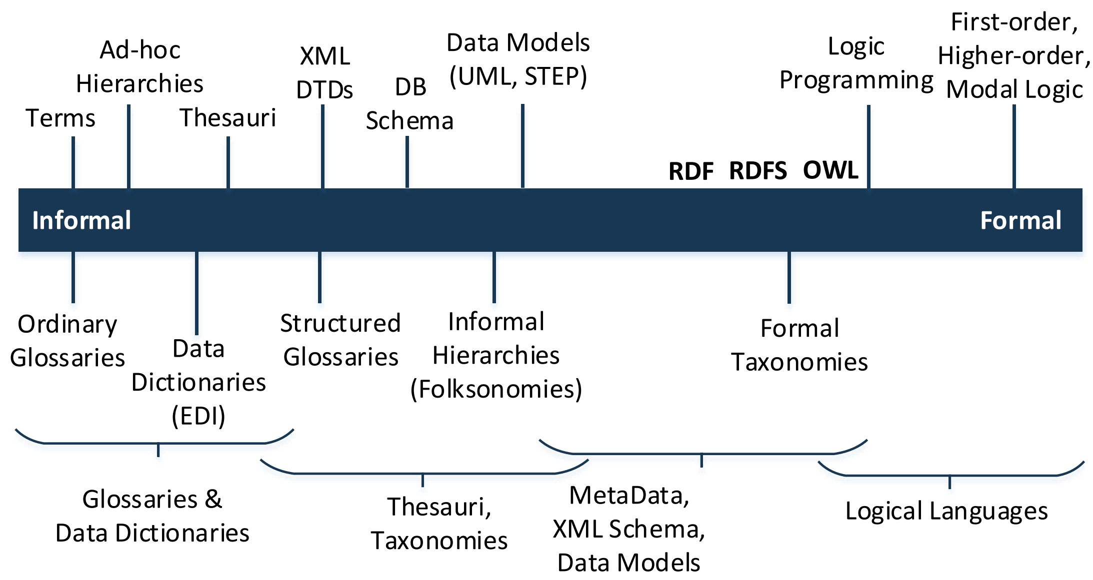


Figure 2.6: **Expressivity of languages –** $\boldsymbol{L}$ (adapted from [37]). The expressivity of languages for the representation of the information is described ranging from Informal and less expressive, e.g., Glossaries, to Formal and more expressive languages , e.g., First-order logic.

Definition 2.2: Ontology [37]

Let C be a conceptualization, and L a logical language with vocabulary V and ontological commitment K. An ontology O_K for C with vocabulary V and ontological commitment K is a logical theory consisting of a set of formulas of L, designed so that the set of its models approximates as well as possible the set of intended models of L according to K.

In practical terms, the trade-off between expressiveness and efficiency when choosing the language L needs to be considered when developing an ontology. Figure 2.6 exposes the differences between existing languages. It is important to note that the difference between the term *vocabulary* and *ontology* is not yet strictly clear. In general, the term ontology is used for more complex and formal collection of terms whereas vocabularies are then referred to as more light-weight ontologies.[9] In this thesis, both terms are used taking into account this observation.

Despite RDF provides an open language to express knowledge it does not make assumptions nor define the semantics about any particular application domain. To define the semantics of a domain a schema for RDF needs to be used, i.e., RDFS. RDFS permits to define a particular vocabulary for RDF data. It specifies the types of objects to which these attributes can be applied. The RDF Schema [38] extends RDF by adding constructs such as `rdfs:Class`, `rdfs:subClassOf`, `rdfs:subPropertyOf`, `rdfs:domain`, `rdfs:range` to mention the most important ones. Important annotations constructs are also added in RDFS such as `rdfs:label` and `rdfs:comment`. For example, the graph in Figure 2.5 can be further extended with these kinds of constructs and

[9] https://www.w3.org/standards/semanticweb/ontology

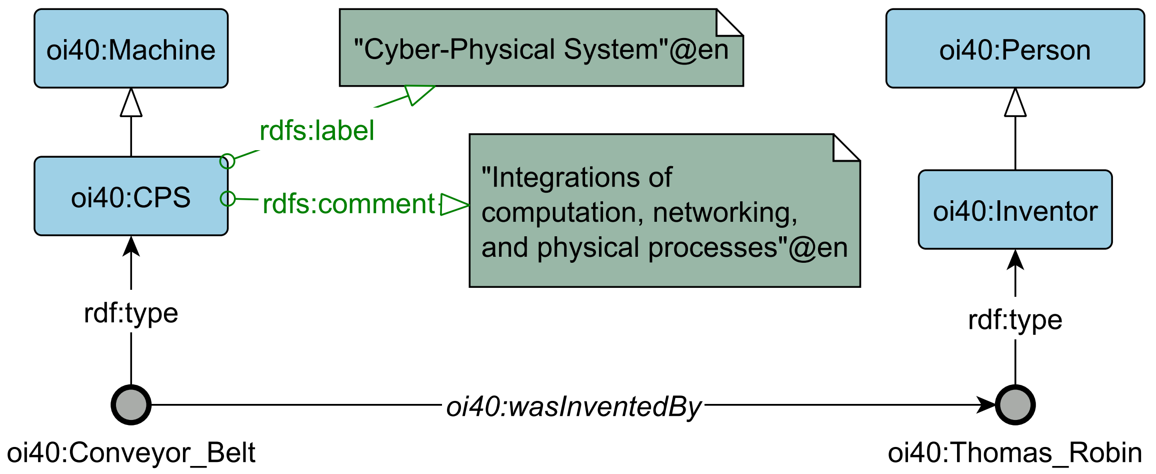


Figure 2.7: **Extended example of a basic RDF graph representing a Conveyor Belt.** Basic inheritance relations and annotation properties representing information about a Conveyor Belt are added to the RDF graph described in Figure 2.5.

annotations to provide meaning to the RDF data.[10] Figure 2.7 shows the definitions of new classes, i.e., `oi40:CPS` is a subclass of `oi40:Machine`. This means that in the domain in which the example is modeled, all CPS are considered as machines. Likewise, the class `oi40:Inventor` is a subclass of `oi40:Person`. It can be also observed from the figure the use of annotation properties such as `rdfs:label` and `rdfs:comment`. In addition, the property `oi40:wasInventedBy` has as its domain, i.e., `rdfs:domain`, the class `oi40:CPS` and as its range, i.e., `rdfs:range`, the class `oi40:Inventor`.

2.2.3 The SPARQL Language

SPARQL W3C[11] is a query language able to retrieve and manipulate data stored in RDF. SPARQL is based on RDF Turtle serialization and graph pattern matching. A graph pattern is an RDF triple containing variables, e.g., subject, property, and object. SPARQL is inspired by SQL, thus many of its features are similar to it. A SPARQL query consists of triple patterns, conjunctions, disjunctions, and optional patterns. Triple patterns are similar to RDF triples where the subject, predicate, and object may be variables.

```
PREFIX oi40: <http://w3id.org/i40/ont/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT  ?cps  ?inventor  ?classes
WHERE  {
?cps  oi40:wasInventedBy  ?inventor .
?cps  rdf:type/rdfs:subClassOf*  ?classes . }
```

Listing 2.2: **Example of a SPARQL query**

In a query, the variables act like placeholders which are bound with RDF terms to build the solutions. Listing 2.2 depicts important features of the SPARQL language. Firstly, it

[10] The rdfs annotation describes the prefix of the RDFS vocabulary: http://www.w3.org/2000/01/rdf-schema#
[11] https://www.w3.org/TR/sparql11-query/

retrieves the instances of the classes which are defined as domain and range of the property `oi40:wasInventedBy`, i.e., `oi40:Conveyor_Belt` as instance of `oi40:CPS` and `oi40:Thomas_Robin` as instance of `oi40:Inventor`. Further, it also infers that the superclass of `oi40:CPS` is `oi40:Machine` by taking advantage of the transitivity feature of the `rdfs:subClassOf` property.

2.2.4 Linked Data

Linked Data is a method for exposing, sharing, and connecting pieces of data using semantic technologies. The main idea behind Linked Data is that the data can be structured using ontologies that describe its meaning. By relying on the collection of semantic technologies such as RDF, OWL, SPARQL, Link Data applications can query that data, and obtain inferences using ontologies. Thus, the machines are capable to understand the connected data. In 2006, Tim Berners-Lee presented four principles for Linked Data [39, 40].

1. Use URIs as names for things.
2. Use HTTP URIs so that people can look up those names.
3. When someone looks up a URI, provide useful information, using the RDF-related standards.
4. Include links to other URIs so that they can discover more things.

The Linked Open Data (LOD) Cloud comprises thousands of datasets and billions of RDF-encoded facts in a report of 2018. Among the largest community-supported Semantic Web knowledge bases are DBpedia [41] and Wikidata [42]. DBpedia is a machine-readable version of Wikipedia,[12] while Wikidata is an envisioned uniform source for Wikipedia articles.

2.2.5 Knowledge Graphs

Currently, the term knowledge graph is a trending term used by big players such as Google, IBM or Microsoft. The term was coined by Google, referring to their use of semantic knowledge in Web Search. It is utilized also to refer to Semantic Web knowledge bases such as DBpedia, Wikidata or YAGO. KGs may employ different knowledge representation formalisms including abstract modeling languages and probabilistic mechanisms not limiting itself to purely RDF. The meaning of the data is stored alongside the data in the graph, in the form of the ontologies. This makes knowledge graphs self-descriptive, a single place to find and understand the data. The semantics of data are explicit and include formalisms for supporting inferencing. KGs comprise large volume of items, and allow for the description of the meaning of their main concepts and relations. They can offer recommendations for how data may need to be adjusted to meet data model requirements. They also enable drawing conclusions and new information from the available data. Further, KGs have proven to be successful to cope with semantic interoperability conflicts during data integration in different domains such as medicine [11], agriculture [12], and human traffic [13]. For factories, KGs are considered to be at the core of the next generation of Enterprise Information Systems [14]. Typically, a KG comprise the following features [43].

1. Mainly describes real-world entities and their interrelations, organized in a graph. Their focus is more on the actual instances (ABox). The schema (TBox) plays a minor role.

[12] http://www.wikipedia.org

2. Defines possible classes and relations of entities in a schema.
3. Allows for potentially interrelating arbitrary entities with each other.
4. Acquires and integrates information into an ontology and applies a reasoner to derive new knowledge [44].

Definition 2.3: Knowledge Graph

Formally, a knowledge graph is defined as a labeled directed graph encoded using the RDF data model [45]. Given sets I and V that correspond to URIs identifying entities in a RDF document and terms from a ontology, respectively; furthermore, let L be a set of literals. A knowledge graph G is a set of triples of the form $(s, p, o) \in I \times V \times (I \cup L)$.

2.3 Data Integration

The main problem of this thesis can be addressed from a Data Integration perspective. Data integration is the process of combining data from diverse sources and providing a unified view to work with them. Data sources comprising semantic heterogeneity conflicts that need to be integrated to provide a unified view are typical in I40 scenarios. Formally, a Data Integration System (DIS) is defined as follows:

Definition 2.4: Data Integration System [46]

A data integration system IS is defined as a tuple $\langle O, S, M \rangle$, where O is the global schema, e.g., RDF Schema, expressed in a language L_O over an alphabet A_O. The alphabet A_O consists of symbols for each element in O. S is the source schema, expressed in a language L_S over an alphabet A_S. The alphabet A_S contains symbols for each element of the sources. M is the mapping between O and S that is represented as assertions: $q_s \to q_o$; $q_o \to q_a$. Where q_s and q_o are two queries of the same arity, q_s is a query expressed in the source schema, q_o is a query expressed in the global schema. The assertions imply correspondence between global and source concepts.

2.3.1 Semantic Interoperability Conflicts

In general, interoperability can be defined as a measure of the degree to which diverse systems, organizations, and/or individuals are able to work together to achieve a common goal [47]. In this work, semantic interoperability is understood as the ability of computer systems to exchange data with unambiguous, shared meaning. Achieving semantic interoperability is complex since semantic interoperability conflicts need to be conciliated. Semantic interoperability conflicts denote differences in the modeling of different and/or equivalent concepts and how these concepts are expressed [5]. In the following, these conflicts are explained [48].

SIC1 – **Structuredness**: this interoperability conflict occurs whenever data sources are described at a different level of structuredness, e.g., structured, semi-structured, and unstructured. Structured data sources are represented using schemas of a particular data/knowledge model, e.g., the relational data model; all the represented entities are described in terms of fixed schema and attributes. Semi-structured data sources are also described using a data/knowledge model, e.g., RDF or XML; however, in contrast to structured data, each modeled entity can be represented using different attributes and a predefined and fixed schema is not required to describe an entity.

SIC2 – **Schematic**: this interoperability conflict exists among data sources that are modeled using a different schema. Conflicts include: i) Different attributes representing the same concept in different sources; ii) the same concept modeled using different structures in the distinct data sources, e.g., attributes versus classes; iii) different types are used to represent the same concept, e.g., string versus integer; iv) the same concept is described at different levels of specialization/generalization; v) different names are used to model the same concept.

SIC3 – **Domain**: this interoperability conflict occurs when various interpretations of the same domain are represented. Different interpretations include: i) Homonym: the same name is used to represent concepts with a different meaning; ii) Synonym: distinct names are used to model the same concept; iii) Acronym: different abbreviations for the same concept are employed.

SIC4 – **Representation**: this interoperability conflict is described when different representations are used to model the same concept. Representation conflicts include: i) Different scales or units; ii) various values of precision; iii) incorrect spellings.

SIC5 – **Language**: this interoperability conflict occurs whenever different languages are used to represent the data or metadata, i.e., schema.

SIC6 – **Granularity**: this interoperability conflict appears when various interpretations of the same domain are represented. Different interpretations include: i) Intra-aggregation: the same data is divided differently, e.g., full person names against first-middle-last; ii) Inter-aggregation: appears when there exist sums or counts as added values.

SIC7 – **Missing Item**: this interoperability conflict occurs whenever different items in distinct data sources are missing. Missing Item comprises: i) Missing attributes; ii) Missing content.

Semantic Interoperability Conflicts in Industry 4.0 Scenarios

In the following, we describe and exemplify some particular semantic interoperability conflicts in I40 scenarios. Three levels of conflicts are identified: i) standardization frameworks; ii) standards; and iii) documents describing a CPS.

Standardization framework related

SIC1 – **Structuredness**: The description of the standardization frameworks is commonly made by means of white papers; thus unstructured sources are used to described standardization frameworks, their layers, levels as well as their relations with standards (cf. [6, 7]).

SIC2 – **Schematic**: The standardization frameworks present schemas for describing functions and standards with the objective of covering I40 scenarios (cf. [6, 49]).

SIC3 – **Domain**: Same standards are classified in distinct dimensions and layers in different standardization frameworks [7].

Standard related

SIC1 – **Structuredness**: Typically, standards are described in unstructured data sources, e.g., PDF documents and excel sheets. The standard *IEC 61360*, an important data dictionary standard for the electro-technical domain[13], can be retrieved as an excel sheet. This standard is typically used in combination with the eCl@ss standard dictionary which is available as an unstructured document, i.e., HTML or PDF. Thus, same concepts are described in these standards and the structure used to represent them is not the same.

SIC2 – **Schematic**: The schemas of the AML and the OPC UA standards are employed to model the same CPS [9, 50].

SIC3 – **Domain**: *Homonym*: same terms are described with different meanings in different standards; e.g., Resource is described in *ISO 15704* as follows: An enterprise entity that provides some or all of the capabilities required by the execution of an enterprise and/or business process; whereas in *ISO 10303* as something that may be described in terms of a behaviour, a capability, or a performance measure that is pertinent to a given process. *Acronym*: different abbreviations are used to refer to the same standard; e.g., IEC 62541 and OPC UA. *Synonym*: distinct names are utilized to express the same meaning, e.g., an *InternalElement* in AML describes the same meaning than an *Object* in OPC UA.

Semantic Heterogeneity Conflicts in CPS Biffl *et al.* [51] and Kovalenko and Euzenat [52] have characterized semantic heterogeneity conflicts in the engineering domain, i.e., CPS-related. The authors have identified the following types of semantic heterogeneity:

M1 – Value processing: same entities are not modeled equally, the relation between values of two entities can be modeled by a function taking a value on one side as an input and returns a value on another side as an output. For example, using different string values, datatypes or mathematical functions; This is an instantiation of the **SIC4** heterogeneity.

M2 – Granularity: same objects are modeled at different levels of detail; This is an instantiation of the **SIC6** heterogeneity.

M3 – Schematic differences: a divergent way of representing semantics for the same object; This is an instantiation of the **SIC2** heterogeneity.

M4 – Conditional mappings: relations between entities exist only if certain conditions occur; This can be seen as **SIC4**.

M5 – Bidirectional mappings: relations between entities have to be defined bidirectionally; This can be interpreted as **SIC4**.

M6 – Grouping and aggregation: different semantic modeling criteria are applied to group elements for the same object; This is an instantiation of the **SIC6**.

M7 – Restrictions on values: mandatory values for properties in the object that have to be handled in the mapping process. This can be seen as **SIC4**.

[13] https://cdd.iec.ch/cdd/iec61360/iec61360.nsf

2.3.2 Semantic Data Integration

In order to comprehend the concept of Semantic Data Integration, an understanding of the definition of semantics is needed. Semantics describe the *meaning* of a word or concept. Firstly, semantics are used to ensure that two concepts, which might appear in different data sources in different forms with different names, can be described as equivalent, i.e., they describe the same entity. The ability to distinguish and conciliate among different semantic heterogeneity conflicts is essential when integrating data from diverse data sources. Secondly, semantics describe the specific form of the relationship that exists between concepts rather than just co-occurrence in text or lexical equivalence of a label. This enables a fully descriptive representation of all of the data available, showing what entities interact with and what role they might have in a given context [53]. Semantic Data Integration is a mechanism which associates different sources of data on the basis of the meaning of data content. This is usually applied to merging the content of different data sources so that an end user may use all the sources through some unified mechanism. This mechanism also tackles the problem of semantic heterogeneity in the data sources.[14] The key to semantically integrate data is the correct management of the meaning of entities in the domain, i.e., the detection and solution of the semantic heterogeneity conflicts in the domain of the data to be integrated.

2.3.3 Rule-based Systems for Semantic Data Integration

There exist many techniques that are utilized for semantic data integration. In the following, rule-based systems techniques that are used during the development of this work are described.

Datalog

One of the techniques used in this work for data integration is Datalog. Datalog is a declarative programming language used to work with deductive databases. Since Datalog rules are a representation of clauses in the function-free Horn fragment of first-order logic (FOL), Datalog revealed itself relevant also for semantic web applications such as ontological modeling and reasoning [54]. A Datalog rule can be expressed as follows:

$$L_0 : -L_1, \ldots, L_n, n \geq 0 \tag{2.1}$$

```
hasRefSemantic(X,T) ∧ hasRefSemantic(Y,Z) ∧  sameRefSemantic(T,Z) ⇒ sameAttribute(X,Y)
```

Listing 2.3: **Example of a Datalog Rule.** The rule represents the semantic equivalence between two elements, i.e., Attributes of the AML standard checking whether the value for their respective semantic references is equivalent.

The atom L_0 is the head while the set of atoms $L_1, ..., L_n$ are called the body. In other terms, a Datalog rule is a function-free Horn clause. In Datalog, every variable in the head of a rule must appear in the body of the rule. A Datalog program is a finite set of Datalog rules. Datalog and OWL can be jointly employed since they share the same interpretations:

- OWL individuals are constants;
- OWL classes are unary predicates; and

[14] https://www.igi-global.com/dictionary/semantic-integration-research-environments/26315

- OWL properties are binary predicates.

For example, the rule in Listing 2.3 describes the fact that two attributes, i.e., X and Y are considered as the same where the respective value of their semantic references, i.e., T and Z, represented by the predicate `hasRefSemantic` are the same. In this case, the `hasRefSemantic` which is a binary predicate, can be seen as an OWL object property connecting two constants, i.e., OWL individuals.

Probabilistic Soft Logic

Probabilistic Soft Logic (PSL) [55, 56] is a framework for collective, probabilistic reasoning which allows defining probabilistic models over continuous variables. The basic building block of PSL are: (1) atoms to model the continuous random variables; (2) predicates which describe relations or properties; and (3) rules combining predicates and atoms to capture dependencies or constraints of the domain based on which it builds a joint probabilistic model over all atoms. Each rule has an associated non-negative weight that captures the relevance of a rule for a given domain. PSL uses soft truth values in the interval [0,1], which allows incorporating similarity functions into the logical model. A PSL model is defined using a set of weighted rules in first-order logic, as follows:

$$\begin{aligned} &Component(A, X) \wedge Component(B, Y) \wedge \\ &SimilarAttributes(X, Y) \Rightarrow Component(A, B) \mid 5.0 \end{aligned} \tag{2.2}$$

PSL utilizes the *Lukasiewicz* t-norm and co-norm to provide relaxation over the logical connectives AND ($\wedge$), OR ($\vee$), and NOT ($\neg$) as follows:

$$\begin{aligned} p \tilde{\wedge} q &= max\{0, p + q - 1\}, \\ p \tilde{\vee} q &= min\{p + q, 1\}, \\ \tilde{\neg} p &= 1 - p \end{aligned}$$

A rule is grounded when substituting constants for variables in the atoms of a rule. For a ground rule $r \equiv r_{body} \rightarrow r_{head} \equiv \tilde{\neg} r_{body} \tilde{\vee} r_{head}$. r_{body} and r_{head} are logical formulas which are composed by atoms and logical operators. The rule r is satisfied (i.e., $I(r) = 1$, iff $I(r_{body}) \leq I(r_{head})$). An *Interpretation (I)* over the atoms in r determines whether r is satisfied, and, if not, its distance to satisfaction. With the *Interpretation (I)*, the rule's distance to satisfaction is defined by the following equation:

$$\phi_r(I) = max\{0, I(r_{body}) - I(r_{head})\} \tag{2.3}$$

An *Interpretation (I)* of a set of ground atoms is a full assignment of soft-truth values to that set. PSL defines the distance to satisfaction for each grounded instance of a rule. For example, assuming the following evidence: $I = Component(A, X) = 0.9, Component(B, Y) = 0.8$ and $SimilarComponent(X, Y) = 0.9$, r is the result of the ground of Rule 2.2. Then, $Component(A, X) \wedge Component(B, Y) \wedge SimilarAttributes(X, Y) = max\{0, 0.9 + 0.8 + 0.9 - 1\}$, i.e., 1.6. The value of the head, $Component(A, B) = 0.8$. Therefore, the distance to satisfaction $\phi_r(I) = max\{0, 1.6 - 0.8\} = 0.8$. In general, a PSL program defines a probability distribution from a logical formulation expressing relationships between continuous random variables. The probability distribution function is as follows:

$$f(I) = \frac{1}{Z} exp \left[- \sum_{r \in R} w_r \phi_r (I)^p \right] \tag{2.4}$$

Where R is the set of ground rules, w_r is the weight of rule r, $p \in \{1, 2\}$ is a modeling parameter which defines whether rules are quadratic or linear and Z the normalization constant. PSL utilizes the most probable explanation inference (MPE). MPE finds the overall interpretation with the maximum probability given a set of evidence. When the value of the probability is the highest, then, the probability of the interpretation is the lowest distance to satisfaction. PSL finds the interpretation that tries to satisfy the rules as much as possible. In this setting, MPE allows to find the interpretation that minimizes $\sum_{r \in R} w_r \phi_r (I)^p$. As recognized in the literature [57], PSL can efficiently and scalable solve this optimization problem.

Expressive and Declarative Ontology Alignment Language

The Expressive and Declarative Ontology Alignment Language (EDOAL) [58] allows for representing correspondences between entities of different ontologies.[15] EDOAL uses classes, relations, properties, and instances constructs to represent ontological entities. Each correspondence models a relationship between the entities, i.e., equivalence, subsumption, disjointness, and membership of an individual to a class [59]. These correspondences are defined as rules. Next, the rules are executed in the Alignment API [60] to obtain the differences and perform the alignment between two ontologies.

2.3.4 Fusion Policies

After identifying equivalent entities in the knowledge graphs, fusion policies are employed to decide how equivalent entities are merged [61]. The fusion policies include: i) **Union:** creates a new entity with the union of the properties of the matched entities; i.e., pairs that are syntactically the same, are unified into a single pair; ii) Subproperty policy. The policy tracks if a property of one RDF molecule is an *rdfs:subPropertyOf* of a property of another RDF molecule; iii) **Semantic based Union:** creates a new entity with the union of the properties of the matched entities; and iv) **Authoritative Merge:** outputs one RDF graph with the data provided from an authoritative source.

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

CHAPTER 3

Related Work

This chapter outlines the state-of-the-art with respect to the work conducted in this thesis. Relevant approaches related to the research problem as well as to the research questions are investigated. Section 3.1 examines general approaches for conducting semantic data integration. Next, in Section 3.2 we review existing methods for representing standards by means of ontologies and knowledge graphs. Section 3.3 overviews existing works for formalizing standards by means of ontologies. Section 3.4 reports similar works in the field of semantically exploiting standards and standardization frameworks using knowledge graphs. Approaches for integrating data of I40 standards are investigated in Section 3.4.2. Finally, in Section 3.5 a review and critical discussion of the current approaches for semantic technologies in manufacturing is carried out.

3.1 Generic Semantic Data Integration Approaches

Generic semantic data integration approaches aim at solving semantic heterogeneity conflicts independently of the domain. Several researchers have tackle the problem of semantic data integration from different views. Ontology-Based Data Integration (OBDI) is one of the most common techniques for solving this problem [62–64]. OBDI approaches are commonly used for semantic data integration since ontologies provide a semantic representation of the domain. In general, the OBDI approach comprises three components: i) the ontology for represent the knowledge of the domain; ii) the data source which typically contain the data of the domain; and iii) the mappings between the two components [62]. Cruz *et al.* [65] discuss different views of the use of ontologies for semantic data integration: i) Single ontology approach. All sources are directly related to a shared global ontology; ii) Multiple ontology approach. Each data source is described by its local ontology separately; and iii) Hybrid ontology approach. A combination of the single ontology approach for describing each data source in the domain with mappings to a general shared ontology. Other studies focus more on the necessary dimensions for developing the mappings. In this regard, three dimensions for mappings are researched: i) the discovery of mappings among ontologies; ii) the declarative formal representation of the mappings; and iii) the reasoning with the mappings. Mappings are required to link two ontologies representing the same domain and comprising semantic heterogeneity conflicts between them [64].

Knoblock *et al.* [66] present KARMA, a semi-automatic framework capable to map structured sources to ontologies to build semantic descriptions of the sources. KARMA allows for the modeling of structured sources. Further, KARMA is able to generate a source model where semantic heterogeneity conflicts between the sources are solved. SILK is a framework for integrating heterogeneous data sources [67]. SILK identifies `owl:sameAs` links among entities of two RDF datasets. This framework also enables the application of data transformations, i.e., data cleaning, data transformation, to structured data sources to perform the integration. Sieve [68] is a framework for assessing the quality of the data to be integrated. Then, Sieve determines which data should be conserved, transformed or discarded. Finally, Sieve applies various fusion policies on top of the data to semantically integrate it. Collarana *et al.* [61] introduce MINTE, an integration framework that collects and integrates data from heterogeneous sources into a knowledge graph. MINTE implements semantic integration techniques that rely on the concept of RDF molecules to represent the meaning of data. This approach also implements fusion policies for merging the RDF molecules and solve semantic heterogeneity conflicts between the heterogeneous sources. Rahm [69] describes the need for a holistic data integration approach capable of scaling to many data sources. The author revises six uses cases where a so-called holistic data integration is applied, i.e., meta-search, open data, integrated ontology, knowledge graphs, entity search engines, and comparison portals. By analyzing these use cases, the author argues that semantic data integration approaches should be performed on the physical level as well as on the use of clustering-based approaches to match entities and metadata (concepts, attributes). Further, a general clustering strategy for entity resolution is proposed with the aim to become a holistic approach that can be used in different domains and use cases. LDIF [70] presents a framework for integrating Linked Data at a large scale. LDIF comprises a mapping language for translating data from the different vocabularies which are used on the Web to a local target vocabulary. To translate the data that is modeled by means of different vocabularies into a local one, LDIF uses the R2R framework. Furthermore, for solving the heterogeneity conflicts LDIF relies on the SILK framework. Finally, this LDIF provides a data quality assessment of the integrated data.

To sum up, the aforementioned approaches for semantic data integration provide generic views to integrate structured data. Still, a lot of manual work is needed for achieving the integration. Most of the revised methods focus on the resolution of semantic interoperability between data sources. Additionally, specific semantic heterogeneity conflicts for standards and standardization frameworks are not considered. On the contrary, in this thesis, we focus on the problem of semantically identifying and integrating equivalent entities in the I40 domain, e.g., standards and standardization frameworks.

3.2 Integrating Industry 4.0 Standards into Knowledge Graphs

Knowledge graphs-based approaches for representing I40 standards are concerned with the use of the semantics of ontologies and knowledge graphs to express the shared knowledge of the standards and solve interoperability conflicts in the domain. Chungoora *et al.* [71] explore the potential of ontology based approaches for representing and exploiting the semantic of the standards in the context of smart manufacturing. Authors propose the use of a heavyweight ontology-based method for representing general features about standards. Hodges *et al.* [72] present an approach for the semantic development and integration of standards towards achieving interoperability between them. This work is of particular interest since i) they recognized the need for the semantic representation of standards by means of ontologies; ii) they identify relevant standards of use for the Industry 4.0 domain and iii) they identify well-known ontologies to be considered in the reuse of new ontologies to represent standards. In this approach, relevant standards for smart manufacturing are identified, the identification of some basic semantic heterogeneity conflicts is performed and a semantic based solution is outlined. Trappey *et al.* [73] also analyze I40 related standards. They focus on the role of CPS for I40 and classify standards according to different levels, e.g., smart connection, data-to-information conversion, cyber-computation, cognition, and configuration. The authors also build a CPS ontology based on the aforementioned concepts as well as on CPS-related terms. Ansari *et al.* [74] introduce an ontology for solving problems in CPS. In this work, the problem solutions and social aspects of CPS in the I40 domain are examined. Human interactions with CPS are considered as a crucial point for the problem solving in the context of the I40 vision. In order to categorize this knowledge an ontology is developed. The ontology covers three types of profiles for describing problem and solutions: 1) Problem-Solving Profile which investigates processes and activities for the solution of problems; 2) Problem-Solver Profile refers to complementarity use of the strengths and weaknesses of humans and CPS with respect to the solution of problems in the I40 domain; and 3) Solution-Profile creates a link between the first two profiles. In [75] the authors discussed how modularization and reuse of ontologies can enhance interoperability in the manufacturing domain. They highlight the needs for semantics across the systems participating in the production life-cycle of manufacturing. Authors refer to existing semantic interoperability conflicts between representations of standards, e.g., IEC 61512. In addition, a set of requirements for ontology developed in this domain are mentioned and a basic procedure for the creation of ontologies is described.

Existing ontology-based approaches for representing I40 standards suffer from several limitations. First, no dedicated ontology is considered for semantically representing standards and standardization frameworks concepts and their associated metadata. Second, relations among standards are identified to some extend but are not modeled by means of an ontology. Third, the examined approaches suffer from the fact that they do not characterize semantic heterogeneity conflicts in the domain as well as no methodological steps are proposed to represent standards by means of ontologies. Contrary, our approach in Chapter 4 presents the development of the Industry 4.0 KG (I40KG). The I40KG is based on the semantic encoded in the standard ontology (*STO*). The *STO* ontology covers the concepts of standards and standardization frameworks as well as the metadata associated with them, which is necessary for representing the knowledge in this domain. Further, relations of standards are semantically described in *STO*.

3.2.1 Solving Semantic Heterogeneity Conflicts among Standards and Standardization Frameworks

Existing works for solving semantic heterogeneity conflicts refer to the identification of standards and their alignment to a level or layer of certain standardization frameworks. Lin *et al.* [8] present similarities and differences between the RAMI4.0 model and the IIRA architecture. Based on the study of these similarities and differences authors proposed a functional alignment among layers in RAMI4.0 with the functional domains and crosscutting functions in IIRA. Additionally, in this work, the IICF framework, which extends IIRA, outlines layers of IoT and identify standards for each one of these layers. Furthermore, the layers in RAMI4.0 are aligned to the IICF layers. For example, while RAMI4.0 specifies OPC UA as the core connectivity standard for connecting manufacturing products, equipment and process software, IICF also specifies OPC UA and adds other three standards, i.e., TCP/UDP/IP, TSN and wireless technologies. Lu *et al.* [22, 76] describe a standardization landscape for smart manufacturing systems. The landscape is built upon relations of standards with products, production systems, and business life-cycle dimensions. The landscape is also described in terms of standards organizations as well as types of standards acting in each of the three dimensions. A framework to analyze the IoT standardization is presented in [77]. In this work, smart manufacturing is considered as a vertical dimension of IoT. A standard database classifying standards is defined in an abstract way, e.g., generic and domain-specific standards. Finally, they identify general gaps of standards and their functions related to IoT. Herzog et at. [78] reported on the needs of semantic-based approach for interoperability in IoT-based automation infrastructures. They provide a comparison among some of the most known architectures for achieving interoperability in IoT domains which are, in practice part of I40 domains. For instance, they include RAMI4.0 and IIRA and define some mappings between these standardization frameworks. They also highlight the necessity of providing a common core information model capable to manage the semantic interoperability conflicts presented in the standardization frameworks.

Siltala [79] investigates existing standards for smart manufacturing and the relations that exist between them. Additionally, a generic model defining concepts such as standards, standard groups, and interfaces is presented. The proposed model has the process concept as a center and relates standards with the processes that they can cover. Li *et al.* [80] describe commonalities and differences between existing reference models for Smart Factories from Germany (RAMI4.0), the US (NIST), and China (MIIT&SAC). Based on this analysis, a framework for smart manufacturing is presented. This framework is focused on the application layers and life-cycle/value streams. Zhao *et al.* [81] propose the use of an open industrial knowledge graph for intelligent manufacturing. The industrial knowledge graph is conceived as a map of connection of domain ontologies and instances. Based on it, a strategy is proposed to solve semantic heterogeneity conflicts at a rather high level. This strategy includes feature matching based on semantic similarity, numeric matching based on rules, and function matching based on task decomposition. Galinski [82] examines the problem of semantic data integration and interoperability among standards. This work describes the need for metadata, data models and metamodels for standards. It also presents an interesting description of which data to consider when describing a standard. Engel et al. [83] present an ontology-based method for automating the engineering of batch process plants. Authors combine domain-specific languages with an ontology. Existing standards for batch processing such as BatchML are revised and combined with ontologies. The method comprises three steps: 1) process recipe that are utilized for modeling process steps and to determine technical requirements; 2) ontological inference which is capable of finding requirements of batch processing plants, e.g., the features of a specific material; and 3) intelligent orchestration. The inferred knowledge obtained in the ontological inference step is used for an orchestration algorithm to combine process modules and finding appropriate engineering solutions. An architecture of three layers, considering an ontology is the top layer, is introduced.

Many shortcomings can be outlined by investigating aforementioned methods. First, their focus is on identifying and classifying existing relations and semantic interoperability conflicts between standardization frameworks, e.g., [8, 22, 76, 80]. Second, the rest of the approaches only target the integration between the information models of standards, e.g., [72, 81]. Conversely, our approach targets to solve semantic interoperability conflicts not only between standards but also among standards and standardization frameworks.

As showed above, the aforementioned approaches comprise several limitations to resolve semantic interoperability conflicts between standards as well as among standards and standardization frameworks. This fact impedes the semantic representation of entities in the domain and negatively impacts the solution of semantic heterogeneity conflicts. To meet this need, **RQ1** is defined. Further, the approach presented in Chapter 4 provides a methodological foundation for the creation and refinement of the I40KG. A characterization of existing semantic heterogeneity conflicts that are common in these scenarios is presented. Also, policies of how to combine heterogeneous information in knowledge graphs are described with a particular example in the area. Finally, the I40KG comprises the semantic metadata of the main concepts in the domain allowing for the identification and solution of semantic heterogeneity conflicts.

3.3 Semantic Representations of Industry 4.0 Standards Using Ontologies

Various approaches investigate the use of ontologies for representing I40 related standards [84, 85]. In this section, we first categorize existing approaches regarding the general applicability or methodology for ontologies representing I40 related standards. Further, related approaches are examined w.r.t. the three ontologies that are developed in the context of this thesis, i.e., the RAMI4.0 ontology (cf. Section 5.2), the AutomationML ontology (cf. Section 5.3), and the SCORVoc vocabulary (cf. Section 5.4).

Sabou *et al.* [86] develop an approach for the semantic modeling and acquisition of engineering knowledge. The authors provide plausible principles that are not only applicable for engineering knowledge but also for the representation of semantics in I40 scenarios. For instance, they examined different ontology engineering methodologies as well as ontologies that can be adapted to these scenarios. In addition, relevant ontology design patterns for engineering are researched in this work. Damjanovic *et al.* [87] present a method for developing ontologies for engineering settings in which different disciplines, i.e., mechanical, electrical, and software are involved in the design of a CPS. The authors investigate the mapping of models representing the divergent disciplines, to the foundational ontology DOLCE. They also introduced the utilization of a dynamic methodology for creating ontologies in these settings. In addition, this methodology incorporates the ODP as a crucial step for guiding domain experts to formalize knowledge.

Szejka *et al.* [88] explore the application of reference ontologies for semantic interoperability in an integrated product development process in smart factories. In this approach, ontologies are proposed for modeling data sources related to the product development and achieve a semantic reconciliation process among entities. Although this approach is interesting, fails to take into account specific semantic heterogeneity conflicts. Additionally, it does not consider the concrete use of ontologies for solving conflicts when creating alignments between entities. Thus, we note here a room for improving semantic representations of standards by means of ontologies as stated in the **RQ2**.

3.3.1 Semantic Representations of Assets in Industry 4.0 Scenarios

As for the semantic representation of assets considering the concept of the Administration Shell as a core, Tantik *et al.* [94] propose an integrated data model and structure for the Administration Shell in I40 contexts. They combine specifications of Authors implemented a use case for the data model. They used a robot arm as a CPS described by the Administration Shell. Pethig *et al.* [95] developed a data model for the Administration Shell to be applied in reconfiguration of conditioning monitoring services for I40. The data model utilizes the properties described in the standard IEC 61360. Diedrich *et al.* [96] investigate a model for semantic interoperability of communication of assets within the smart factories context. This work depicts the principles for an interaction model of components described by the Administration Shell. Specific mappings of

Table 3.1: **Comparison of asset representations in manufacturing environments.** Different approaches aiming to represent entities in the context of smart manufacturing are compared according to criteria such as: basic concepts, identification, data model, organization type, and serialization.

Approach	Basic Concept	Identification	Data model	Organization Type	Formalization
EDDL [89]	Device	n/a	Object	n/a	Text
OMM [90]	Physical artifact	Primary ID and IDs for blocks	Element	Hierarchical	XML
DOMe [91]	Object	Primary ID and IDs for blocks	Object	Hierarchical	XML
PML [92]	Physical object	XML tag ID	Object	Hierarchical	XML
SPDO [93]	Product	URI/IRI	Resource	Hierarchical	OWL-DL
RDF-Based Approach	Administration Shell/Asset	URI/IRI	Resource	Hierarchical	RDF, RDFS, OWL

the data model to important standards in the domain to describe a CPS, e.g., AML and OPC UA are included as a proof of concept. The *Electronic Device Description Language* (EDDL) is a language to delineate information related to digital components [89, 97]. EDDL is available for a large number of devices that are currently utilized in the process industry. EDDL provides a text-based description of devices and their properties, describing the data and how they are displayed. The *Object Memory Model* (OMM) is an XML-based format that allows for modeling information about individual physical artifacts [90]. The memory is partitioned into blocks to enable various actors to read and write different aspects of information about an artifact. The conceptual approach in that work is to bring a semantic layer to the physical components, but its implementation suffers from the syntactic limitations of XML. However, it is envisioned that blocks of an OMM contain RDF and OWL payload data. Extending the concept of OMM, *Domeman* [98] is a framework for the representation, management, and utilization of digital object memories.

The idea of using semantic descriptions of physical artifacts by combining OMM and a server realization has been proposed by Haupert and Schneide [99]. The authors developed an object memory server as an index server for product memories, based on the same set of metadata as the block format. However, this approach is focused on the identification of artifacts and still exposes the OMM limitations mentioned above. A similar approach is proposed with *DOMe* in [91]. DOMe is a Digital Object Memory which allows automated interaction between workpieces and machine tools using an RFID-based smart environment. It also relies on the metadata presented by the OMM approach to describe the manufacturing object. The application of ontologies is considered for representing rules of the manufacturing domain. However, the semantic description of the object itself, and the various types of data that exist in the manufacturing domain, are not addressed. The *Physical Markup Language* (PML) is a common language for describing physical objects, processes and environments [92]. The goal of PML is to use these descriptions in remote monitoring and control of the physical environment. Janzen *et al.* [93] define smart products as a connection of physical products and information goods that allow the embedding of digital product information into physical products. They present the *Smart Product Description Object* (SPDO), a data model built on top of the *DOLCE* ontology for describing smart products. Bergweiler [100] defines an approach for distinguishing local and global data structures stored in Active Digital Object Memories (ADOMe), to extend so-called smart labels with memory and processing capabilities. According to the author, this can be realized by storing the data in a unified structured format.

Table 3.2: **Comparison of semantic representations of the AML standard with AMLO**. Comparison of existing AutomationML ontologies with AMLO; The comparison is performed regarding the use of ontology engineering methodology, the inclusion of ODPs, the reuse of terms from other ontologies, availability of ontology sources, the source used to develop the ontology, the formalization as well as the generality of the use case in which the ontology is utilized.

Approach	**Methodology**	**ODPs**	**Reuse**	**Availability**	**Source**	**Formalization**	**Use Case**
Abele *et al.* [101]	No	No	No	-	CAEX	OWL	Plant Validation
Björkelund *et al.* [102]	No	-	-	-	-	-	Skill Representation
Glawe *et al.* [104]	No	No	No	-	-	-	Automation Security
Persson *et al.* [103]	No	No	No	-	-	-	Knowledge Integration
Hua *et al.* [105]	No	No	Yes	-	-	-	Model-driven for Robotics
AML Ontology	Yes	Yes	Yes	Yes	CAEX	OWL	General

In summary, the investigated works for representing the semantics of an asset in industry contexts, particularly by considering the Administration Shell, suffer from many limitations. Table 3.1 provides a comparison of existing ontological representations of the Administration Shell concept as a part of RAMI4.0. We considered all these limitations and propose an ontology for representing the RAMI4.0 model as well as the Administration Shell concept from a semantic view in Section 5.2.

3.3.2 Semantic Representations of the AML Standard

In recent years, much attention has been paid to represent the knowledge regarding the automation domain by using ontologies [87]. Concretely, the AML standard has been at the core of many efforts in this regard. The main focus of these works has been in formalizing the CAEX format into an ontology. Abele *et al.* [101] present an ontology for the validation of plant models, e.g., attribute consistency checking and correctness of internal links. The ontology covers base concepts of AML and how they are mapped to OWL; Björkelund *et al.* [102] model an ontology exploiting core concepts of AML and utilize the resulting ontology as a common vocabulary to transform AML models into RDF; Persson *et al.* [103] describe a knowledge integration framework for robotics. In this context, the knowledge is represented in AML and transformed into RDF to publish the RDF data according to Linked-Data principles. To this end, they propose an ontology covering main AML concepts; Another definition of a AML ontology is developed in the context of using knowledge to support the engineering process of automation systems [104]. While the focus of this work is on security, core concepts of the AML are designed as an OWL ontology. Further, SWRL rules are introduced to logically connect AML elements. Hua *et al.* [105] developed a semantic-based approach to software engineering of industrial robotics. Authors propose an approach to deal with robotic components and how they can be classified and modeled with AML. Further, how AML models can be processed by means of an ontology is demonstrated in this work. To this end, an AML ontology covering main aspects of AutomationML is created.

To analyze existing works for an AutomationML ontology, aspects considered of relevance for ontology development are investigated (cf. Table 3.2), namely: a) The utilization of an Ontology Engineering methodology; b) the inclusion of ODPs; c) the reuse of well-known ontologies; d) the

Table 3.3: **Comparison of semantic representations of SCOR.** The comparison is performed regarding the use of ontology engineering methodology, the reuse of terms from other ontologies, availability of ontology sources, the completeness of the model, the metric structure as well as whether the ontology has been evaluated.

Approach	Methodology	Reuse	Availability	Completeness	Metric Structure	Evaluation
Ye *et al.* [106]	No	No	No	n/a	n/a	No
Fayez *et al.* [107]	No	No	No	Assumed	Hierarchical	No
Leukel *et al.* [108]	No	No	No	No	Hierarchical	No
Sakka *et al.* [109]	No	No	No	N/a	Hierarchical	No
Zdravkovic *et al.* [110]	Yes	No	No	Assumed	n/a	No
Lu *et al.* [111]	No	No	No	No	Hierarchical	No
SCORVoc	Yes	Yes	Yes	Yes	Queries and Properties	Yes

availability of ontology resources, e.g., on Linked Open Vocabularies (LOV);[1] e) the language used as an input for the ontology, e.g., CAEX; f) the language utilized to describe the ontology, e.g., OWL, and, finally g) the main use case in which the ontology is used.

Overall, existing works lack common desirable features for an AML ontology. First, previous ontologies are tailored for specific use cases and do not consider all the details of the AML standard. Second, most of the existing ontologies are designed without considering any methodology for ontology design or best practices such as the inclusion of ontology design patterns or reusing well-known vocabularies. Lastly, besides being described in articles existing ontologies are not available for consulting or reusing. We considered all these limitations and proposed an ontology for representing the AML standard in Section 5.3.

3.3.3 Semantic Representations of SCOR

Various projects and activities aim at describing SCOR into an ontology using RDFS and OWL [107–111]. The formalization of the SCOR model into an ontology is first addressed in [108]. The authors analyzed the different conceptualization levels of the model and converted them into OWL classes. In [110], a seminal approach formalizes Supply Chain operations overcoming the semantic weaknesses of the SCOR model. In this work the SCOR-KOS OWL model is presented, which encodes the main entities and properties for SCOR. In addition, the SCOR-Full ontology is a domain ontology for the representation and management of knowledge about Supply Chain operations. The latter presents the core concepts of Supply Chain embedded in SCOR definitions. This effort is also the basis used by Zdravković et al. [112] to configure the Supply Chain process. They provide a thread model configuring a specific flow of the Supply Chain studied.

The combination of the SCOR ontology and the *ONTO-PDM*[2] ontology is addressed in [111]. The ONTO-PDM ontology is used to represent information regarding product development, which is not covered by SCOR. The goal is to create a Supply Chain ontology framework for networked enterprises interoperability. Sakka et al. [109] present a SCOR model as a way to align the business processes with strategical objectives for Supply Chains. Concepts like information and input/output are included to face this alignment. SCOR is modeled using ARIS[3] thus

[1] http://lov.okfn.org/dataset/lov
[2] ONTO-PDM is an ontology for Product Data Management interoperability within manufacturing process environment, presented in [113]
[3] Business Modeling Approach

obtaining a SCOR/ARIS model. Then, XLST transforms the output of SCOR/ARIS into a SCOR OWL ontology. The work conducted by [114] provides an ontology model to support Supply Chain process modeling and analysis based on the SCOR model. In this work, only part of the SCOR-KOS model [110] related to the definition of input and output entities in SCOR processes is reused.

To analyze existing works for a SCOR ontology, significant aspects for this work as well as for ontology development are examined (cf. Table 3.3), namely: a) The utilization of an Ontology Engineering methodology; b) the reuse of well-known ontologies; c) the availability of ontology resources, e.g., on Github; d) Most approaches choose to stay close to the hierarchical structure for processes and metrics given by the original source; e) Finally, none of these vocabularies are 'operationalized', and enabled to automatically compute KPIs using data.

3.4 Knowledge Graphs for Integrating Industry 4.0 Standards

Integrating data of CPS is of core significance for the development of the I4.0 vision. Recently, there has been a large amount of research investigating the integration of CPS [115], as well as the recognition of the relevance of the semantic technologies in this area [116]. In this section, we describe the state of the art with regard to the integration of CPS. It is important to note that the integration of CPS is performed in a multi-disciplinary environment where different disciplines collaborate. Existing approaches are critically reviewed and classified into two main categories: 1) semi-automatic integration of CPS. These works do not consider the use of ontologies or knowledge graphs for describing the domain or performing the integration; and 2) ontology-based integration approaches for integrating I40 standards.

3.4.1 Semi-Automatic Integration of CPS

In [117], a tool to map two documents describing a CPS is presented. The CPS documents are described in AML. This work allows for the integration of AML documents, their respective descriptions, and the modified parts of one document into the other. Further, a mapping algorithm for AML documents is presented. Nevertheless, the mapping process is performed manually. Himmler [118] presents a framework to create standardized application interfaces in plant engineering based on AML. The work provides a function-based based standardization framework for the plant engineering domain. BI *et al.* [119] present *MSCIM*, a Mechatronics System Common Information Model to support the multi-disciplinary CPS-design. MSCIM relies on XML and XML Web services technologies to leverage the integration. Further, MSCIM utilizes the wrapper integration approach in a very generic level. Lüder *et al.* [120] describe a manual approach for the CPS information integration by means of AML. In this work, different types of information which are typical in CPS design are outlined. Interoperability conflicts occurring between AML documents with different information types are mentioned. Further, a manual approach based on AML is described to integrate AML documents comprising the mentioned types of information. Chen *et al.* [121] develop a framework for the integration of the design of CPS; requirements for each one of the disciplines involved are characterized, as well as the representation of constraints among disciplines. In [122], a method for integrating mechatronic objects design is proposed. This method combines advantages of bottom and top down approaches into a hybrid approach. Bihani *et al.* [123] introduce an automatic technique for integrating documents of CPS design described by different views in the AML standard. The technique describes a middleware concept for the digitalization of workflows, which provides electronic data exchange between independent engineering tools based on the AML standard. In this technique, no consideration of the semantics described in the different views is employed. All these approaches have the potential to solve specific integration problems for CPS. However, they suffer from the limitation of not considering the semantics encoded in the different views produced by the different disciplines involved in the CPS design.

3.4.2 Ontology-based Integration Approaches for Integrating I40 Standards

Ekaputra *et al.* [124] surveyed state-of-the-art approaches for multi-disciplinary engineering environments of CPS design. Table 3.4 reports on existing methods for semantically integrating data when different disciplines are collaborating for CPS creation and design. In this work, a set of criteria for performing the data integration in these environments is derived, i.e., (i) Ontology Language and Framework, (ii) Data Acquisition, (iii) Mapping and Transformation, and (iv) Storage and OBDI data access. With respect to which kind of variant for the OBDI is used authors considered from using a single ontology, multiple ontologies, a hybrid approach combining both and, finally a Global-as-View (GAV) OBDI. The GAV OBDI method provides a unified view of a global ontology. Next, the languages and frameworks used for the integration are presented. The use of RDF and OWL are quite common. However, a few approaches do not use such standards arguing that the level of expressiveness that they can achieve, e.g., in F-Logic is higher than with OWL and SWRL, e.g., Angele *et al.* [134]. Other languages such as XML Topic Maps, e.g., Lee *et al.* [129] and Common Logics, e.g., Imran *et al.* [142] are also employed for these integration solutions.

As for the data acquisition methods, they included the ETL (Extract, Transform, and Load) where particularized transformations of data are developed. Further, the ELT method (Extract, Load, and Transform), which may include transformations to an ontology. The Ontology Based Data Access (OBDA) is an important approach that permit to access data sources, typically relational databases as a virtual RDF graph, e.g., KARMA [66], Ontop [145], or D2RQ [146]. These approach can also manage accessing other type of unstructured data, e.g., CSV, excel sheets. The majority of the surveyed methods apply RDF property matching and URI and Global Unique Identifier (GUID). URI and GUID are used to link equivalent instances from different ontologies. The RDF property matching exploits RDFS and OWL properties for creating the mappings between distinct ontologies, i.e., `rdfs:subClassOf`, `rdfs:subPropertyOf`, `owl:sameAs`, `owl:equivalentClass`. For the definition of mappings, applications such as SILK [67] are employed. Also, taking the advantages of the RDF, RDFS and OWL properties, SPARQL construct queries are employed to create the mappings. In the majority of the cases an add-hoc code is implemented. Interestingly, reasoners and rule engines are included in a few methods as an option for the needed transformations.

With regards to the storage of data, the subcategories of RDF triple store, relational databases (RDBMS) are outlined. For data access, SPARQL endpoints are the most common alternative followed by customized APIs and customized GUIs. Interestingly, stream data engine are also used for accessing data in this types of integration solutions They built a decision tree with the aim to support the choice of selecting an approach depending on parameters such as the level of semantic heterogeneity, mapping complexity, and dynamics of data sources.

Kovalenko and Euzenat [147] investigated ontology matching techniques to execute identification and integration of data in this context. A survey of existing languages for realizing this task is presented; furthermore, EDOAL is proposed for tackling the problem of matching entities for the resolution of semantic heterogeneity conflicts between CPS documents.

Table 3.4: **Technology options for OBDI.** Approaches for OBDI w.r.t. existing technologies and their adoptions for integrating data in multi-disciplinary environments for CPS design (adapted from [124]).

		OBDI Approaches
OBDI Variant	**Single Ontology**	Abele *et al.* [125], Brecher *et al.* [126], Graube *et al.* [127], Hennig *et al.* [128], Lee *et al.* [129], Novak *et al.* [130], Paneto *et al.* [84], Sabou *et al.* [131], Softic *et al.* [132], Terkaj *et al.* [133]
	Multiple-ontology	Angele *et al.* [134], Feldmann *et al.* [135], Kovalenko *et al.* [136], Kharlamov *et al.* [137]
	Hybrid	Arnio *et al.* [138], Dibowski *et al.* [139]
	GAV OBDI	Dubinin *et al.* [140], Ekaputra [141], Imran *et al.* [142], Lin *et al.* [143]
Language and Framework	**RDF**	Arnio *et al.* [138], Abele *et al.* [125], Brecher *et al.* [126], Dubinin *et al.* [140], Ekaputra [141], Feldmann *et al.* [135], Graube *et al.* [127], Hennig *et al.* [128], Lin *et al.* [143], Persson *et al.* [144], Kovalenko *et al.* [136], Paneto *et al.* [84], Kharlamov *et al.* [137], Sabou *et al.* [131], Softic *et al.* [132], Terkaj *et al.* [133]
	OWL	Arnio *et al.* [138], Brecher *et al.* [126], Dubinin *et al.* [140], Ekaputra [141], Feldmann *et al.* [135], Graube *et al.* [127], Kovalenko *et al.* [136], Lin *et al.* [143], Novak *et al.* [130], Persson *et al.* [144], Sabou *et al.* [131], Softic *et al.* [132], Terkaj *et al.* [133]
	OWL2	Dibowski *et al.* [139], Hennig *et al.* [128], Paneto *et al.* [84], Kharlamov *et al.* [137]
	F-Logic	Angele *et al.* [134]
	Topic Maps	Lee *et al.* [129]
	Common Logic (CL)	Imran *et al.* [142]
Data Acquisition	**ETL**	Abele *et al.* [125], Dibowski *et al.* [139], Feldmann *et al.* [135], Lee *et al.* [129], Novak *et al.* [130], Persson *et al.* [144], Softic *et al.* [132], Terkaj *et al.* [133]
	ELT	Arnio *et al.* [138], Ekaputra [141], Kovalenko *et al.* [136], Sabou *et al.* [131]
	OBDA	Kharlamov *et al.* [137]
	Manual	Angele *et al.* [134], Hennig *et al.* [128], Imran *et al.* [142], Paneto *et al.* [84]
Mapping	**RDF Property**	Arnio *et al.* [138], Feldmann *et al.* [135], Lin *et al.* [143], Kovalenko *et al.* [136], Kharlamov *et al.* [137]
	URI/GUID Matching	Abele *et al.* [125], Brecher *et al.* [126], Dibowski *et al.* [139], Ekaputra [141], Graube *et al.* [127], Hennig *et al.* [128], Novak *et al.* [130], Sabou *et al.* [131]
	Property Value Matching	Angele *et al.* [134], Dubinin *et al.* [140], Paneto *et al.* [84]
Transformation	**SILK**	Arnio *et al.* [138]
	SPARQL Construct	Ekaputra [141], Persson *et al.* [144],
	Code	Angele *et al.* [134], Brecher *et al.* [126], Dibowski *et al.* [139], Ekaputra [141], Feldmann *et al.* [135], Graube *et al.* [127], Hennig *et al.* [128], Imran *et al.* [142], Kovalenko *et al.* [136], Lee *et al.* [129], Lin *et al.* [143], Novak *et al.* [130], Sabou *et al.* [131], Terkaj *et al.* [133]
	Reasoner/Rule Engine	Angele *et al.* [134], Dubinin *et al.* [140], Hennig *et al.* [128], Paneto *et al.* [84]
Data Storage	**Triplestore**	Arnio *et al.* [138], Abele *et al.* [125], Dibowski *et al.* [139], Feldmann *et al.* [135], Graube *et al.* [127], Kovalenko *et al.* [136], Paneto *et al.* [84], Persson *et al.* [144], Sabou *et al.* [131], Softic *et al.* [132], Terkaj *et al.* [133]
	In-memory/ file-based	Dubinin *et al.* [140], Ekaputra [141], Hennig *et al.* [128], Imran *et al.* [142], Lin *et al.* [143], Novak *et al.* [130]
	RDBMS	Kharlamov *et al.* [137]
	Others	Abele *et al.* [125]
Data Access	**SPARQL Endpoints**	Arnio *et al.* [138], Abele *et al.* [125], Dibowski *et al.* [139], Feldmann *et al.* [135], Graube *et al.* [127], Kharlamov *et al.* [137], Kovalenko *et al.* [136], Paneto *et al.* [84], Persson *et al.* [144], Sabou *et al.* [131]
	Custom APIs	Abele *et al.* [125], Brecher *et al.* [126], Ekaputra [141], Imran *et al.* [142], Lee *et al.* [129], Novak *et al.* [130], Terkaj *et al.* [133]
	Custom GUIs	Kharlamov *et al.* [137], Softic *et al.* [132], Terkaj *et al.* [133]
	Stream Data Engine	Graube *et al.* [127]

The investigated approaches have the potential to solve specific integration problems for CPS documents in multi-disciplinary settings. However, isolated problems are tackled, and general methods capable of producing a final CPS integrate design considering the identification and solution of semantic interoperability conflicts are still missing. We particularly noted a lack of categorization of existing semantic interoperability conflicts that typically occur in the domain. Additionally, just a few approaches rely on the capabilities of the reasoners and rule engines to perform transformations while validating and discovering domain knowledge. Therefore, novel approaches for integrating CPS while solving semantic heterogeneity conflicts have to be developed as defined by **RQ3**.

3.5 Applications of Semantic Technologies for Data Integration in Factories

In this section, we give an overview on the development and usage of semantic technologies, i.e., knowledge graphs and ontologies, in industrial scenarios.

Adams *et al.* [148] presented a work developed by the Boeing company for the semantic data integration of heterogeneous data sources, i.e. personnel, aircraft maintenance, and training. Authors present a knowledge-based system where ontologies are used to describe the data sources. Further, an upper ontology is developed to represent general terms that are commonly used across all data sources. The Cyc ontology is employed as a basic language to develop the upper ontology. Some basic recognition of semantic heterogeneity conflicts between the data sources is made. The main target of this solution is to support decision making in the aircraft domain.

Siemens developed an *ontology-based access* to their wind turbine stream data [149, 150]. The ontology serves as a global view over databases with different schemata. It thus enables SPARQL queries to be executed on different databases without taking into account different schemas. Another effort driven by Siemens is the creation of their industrial knowledge graph [151]. In this work, a knowledge graph approach is presented to cope with the problem of isolated manufacturing data and the need for an integrated view on top of these data. The domain knowledge of the company is captured in the knowledge graph with the vision of integrated intelligence across the company. Angele *et al.* [134] devise a method for integrating data from different sources at the Software AG. This method relies on F-logic rules for describing the mappings between objects residing in the data sources and ontologies. They consider a business ontology as the top ontology describing all objects in their domain. Further, data sources ontologies are employed for the particular description of the data sources. Two types of mappings, manual and automatic are utilized. Gianfranco *et al.* [152] present an approach for the applicability of Semantic Web technologies in industrial context to enhance semantic interoperability. Authors propose a systematic approach for supporting the development of a semantic model. The development of the semantic model focus on the combination of the reusing existing reference models and the migration of the of existing legacy systems into a semantic representation. Some use cases are presented demonstrating the need for such an approach for smart manufacturing companies. Schabus *et at.* [153] investigate a use case driven approach for semantically annotate data in I40 domains. The objective of this approach is to integrate data from different sources, i.e., spatial and manufacturing data. Two use cases are presented. Of particular interest is the second use case which aims at supporting the decision making in manufacturing scenarios. In this use case authors identify bottlenecks in the semiconductor production line by means of semantic technologies. To achieve the data integration an ontology is developed and the data sources are annotated. Then, all data is migrated into a graph database. Statoil ASA also established a "single point of semantic data access" through an ontology-based data integration for oil and gas well exploration and production data [154]. They thus reduced the time-consuming data gathering task for their analysts by hiding the schema-level complexity of their databases. Ford Motor Company captures knowledge about manufacturing processes in an ontology such that their own developed AI system is able to "manage process planning for vehicle assembly" [155]. Furthermore, Ford examined the potential of federated ontologies to support reasoning in industry [156] as well as detecting supply chain risks [157]. Volkswagen developed a *Volkswagen Sales Ontology*[4] to provide the basis for a contextual search engine [158]. Renault developed an ontology to capture the performance of automotive design projects [159]. With regard to OBDA, Statoil chose the *Ontop* [160] framework because of its efficient query processing. While Siemens initially favored Ontop as well, they developed their own system in the end to further optimize stream data processing. Regarding semantic models for companies, none of the existing works has specifically addressed machine tools and factory infrastructures. While it is understandable that companies prefer not to share internal details of their methodologies and infrastructure, there is nevertheless very limited evidence of semantic technologies being deployed in the manufacturing industry. Therefore, as defined in **RQ4**, there is clear room for improvement on the application of semantic technologies in manufacturing companies to resolve semantic data integration problems in I40 scenarios [161].

[4] http://www.volkswagen.co.uk/vocabularies/vvo/ns

CHAPTER 4

Integrating Industry 4.0 Standards into a Knowledge Graph

Realizing smart factories according to the Industry 4.0 (I40) vision requires intelligent human-to-machine and machine-to-machine *communication*. In order to achieve this goal, components such as actuators, sensors, and CPS along with their data, need to be described; moreover, interoperability conflicts arisen from various semantic representations of these components demand also solutions. With the aim of *empowering communication* in smart factories, a variety of standards and standardization frameworks have been proposed. These standards enable the description of the main properties of components, systems, and processes, as well as interactions among them. Standardization frameworks classify, align, and integrate industrial standards according to their purposes and features. Various standardization frameworks have been proposed all over the world by industrial communities, e.g., RAMI4.0 or IICF. While being expressive to categorize existing standards, standardization frameworks may present divergent interpretations or classifications of the same standard. Mismatches between classifications of standards generate semantic interoperability conflicts which negatively impact the effectiveness of communication in smart factories.

In previous chapters, the research problem, challenges, and research questions are outlined. This chapter addresses the first level of the challenges (cf. Figure 4.2). The semantic interoperability conflicts that occur between the representation of standards made by divergent standardization frameworks. Figure 4.1 shows different representations of the same standard in two standardization frameworks. Hence, there exist a Representation conflict, i.e., **SIC4**. In this scenario, other conflicts are present among standards as well as among standards and standardization frameworks, e.g., **SC1**, **SC2**.

Problem statement. In this chapter, we investigate the problem of semantic interoperability among standards and standardization frameworks in the context of I40. We also identify the main semantic interoperability conflicts and examine the conditions for resolving them.

The following research question is investigated in this chapter:

> **RQ1**: How can a knowledge graph approach define mappings of standards and standardization frameworks and resolve existing semantic interoperability conflicts among them?

In this level, semantic interoperability conflicts for standardization frameworks are investigated. The chapter is based on the following publications [162, 163].

(a) RAMI4.0 IT layer

(b) IICF Model

Figure 4.1: **Standardization frameworks aligned with I40 Standards**. (a) RAMI4.0 IT (adjusted from [20]). (b) IICF Model (adjusted from [8]). OPC UA standardizes machine-to-machine communication, and it is positioned at different levels in RAMI4.0 IT and IICF. OPC UA is at the Framework level in IICF, while in RAMI4.0 IT, it is positioned at communication level and presented as a standard for the description of data management and analytic processes. Thus, the same standard is categorized differently from two standardization frameworks that target the domain of Industry 4.0.

Proposed solution. In this chapter, we tackle the problem of standard interoperability across different standardization frameworks and devise a knowledge-driven approach that allows for the description of standards and standardization frameworks into an Industry 4.0 knowledge graph (I40KG). The *STO* ontology is used for representing the main properties of standards and standardization frameworks, as well as relationships among them. The I40KG integrates more than 200 standards and four standardization frameworks. To populate the I40KG, the landscape of standards in the I40 domain has been surveyed and analyzed from a semantic perspective. The resulting I40KG represents knowledge expressed in more than 200 industrial related documents that include technical reports, research articles, and white papers. Additionally, the I40KG has been linked to existing knowledge graphs and an automated reasoning has been implemented to reveal implicit relations between standards as well as mappings across standardization frameworks. We analyze the number of discovered relations

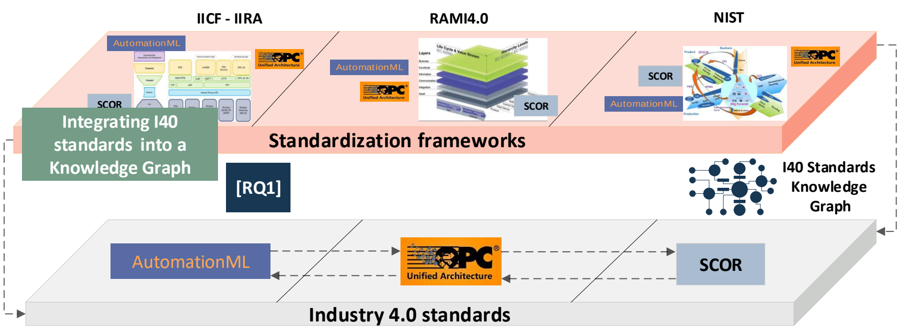


Figure 4.2: **The levels of the research problem addressed in this chapter.** We address challenges concerning the identification of mappings among standards and standardization frameworks.

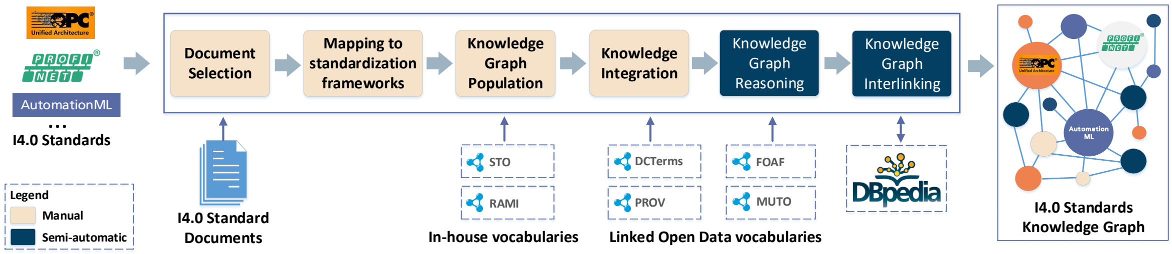


Figure 4.3: **Methodology for creating the I40KG**. Standardization frameworks and their classification of standards are received as input; the output is a graph representing relations among standardization frameworks and standards as well as between standards. *STO* and existing ontologies are utilized to describe existing relations. A reasoning process exploits the semantics encoded in *STO* to infer new relations between standards. The linking with external knowledge graphs, e.g., *DBpedia*, permits the enrichment of the I40KG.

between standards and the accuracy of these relations. The observed results indicate that both reasoning and linking processes enable for increasing the connectivity in the knowledge graph by up to 80%, whilst up to 96% of the relations can be validated. These outcomes suggest that integrating standards and standardization frameworks into the I40KG enable the resolution of semantic interoperability conflicts, empowering thus the *communication* in smart factories. In the following, the contributions of this chapter that particularly assess **RQ2** are outlined.

- A methodology to collect and integrate knowledge about standards and standardization frameworks in a knowledge graph.
- The *STO* ontology to describe standards and standardization frameworks as well as their relationships.
- A knowledge graph for Industry 4.0 (I40KG), containing the semantic descriptions for more than 200 standards and more than 25 standard organizations.
- An empirical evaluation of the quality and accuracy of the integration techniques followed during the creation of I40KG. The observed results provide evidence of the soundness of the discovered relations explicitly represented in I40KG; up to 96% of the discovered relations are valid, and the connectivity is increased by up to 80%.

The remainder of this chapter is structured as follows. Section 4.1 outlines the methodology to create the I40KG as well as some of the design decisions to develop the *STO* ontology. An empirical evaluation of our approach is presented in Section 4.2, while Section 4.3 includes a discussion of the outcomes. Finally, in Section 4.4 the concluding remarks of the chapter are presented.

4.1 I40 Knowledge Graph Creation

This section presents a methodology for the creation of the I40KG. This methodology is composed of five steps: i) Extract Information of Standards; ii) Knowledge Graph Population; iii) Knowledge Graph Integration; iv) Knowledge Graph Reasoning; and v) Knowledge Graph Interlinking (cf. Figure 4.3). In the following, the Standard Ontology, which is used along the methodology, is described. Next, the steps of the methodology are described in detail.

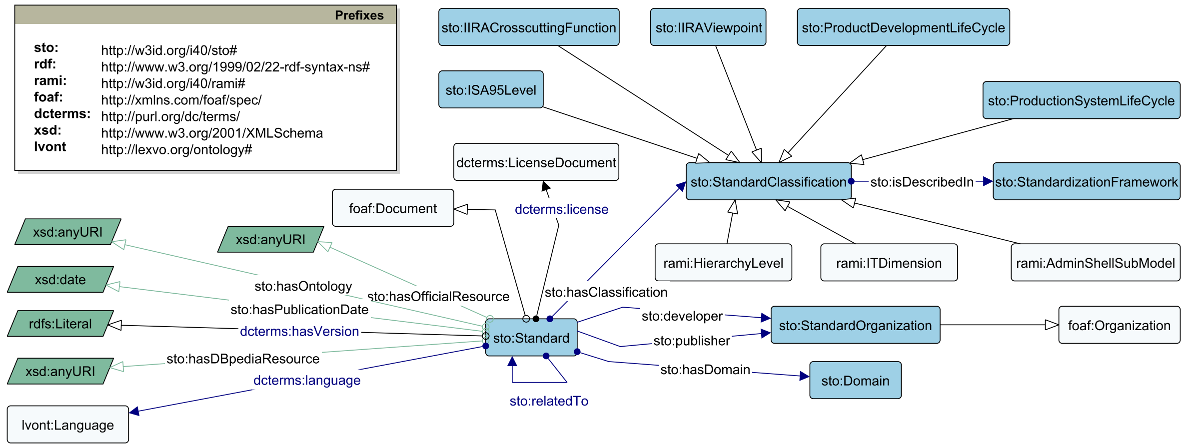


Figure 4.4: **Core classes and properties of the Standard Ontology (*STO*)**. Classes of *STO* are depicted in blue. White classes represent reused classes from FOAF, DCTERMS, and RAMI4.0 ontologies. Reused properties are drawn in blue, e.g., `dcterms:language`. Green rectangles depict datatype properties. Classes and properties are used to describe standards and standardization frameworks in I40 scenarios.

4.1.1 The Standards Ontology

The Standards Ontology (*STO*), is designed to semantically describe standards related to I40 as well as their relations. In addition, main standardization frameworks for I40 and the classification of standards made in them are encoded in *STO*. The development of *STO* follows the methodology for building ontologies of industry related standards proposed in Chapter 5. In this regard, classes and properties from well-known ontologies are reused, e.g., *PROV* for describing the provenance of entities, *FOAF* for representing and linking documents and agents, e.g., persons, organizations, *DCTERMS* for documenting metadata, such as licenses, as well as the *RAMI4.0* ontology for linking standards with *RAMI4.0* concepts. Additionally, the ontology *Lexvo* is employed for linking to the available language of the standard document, e.g., English, German. *VoCol* is used as an integrated environment for ontology development based on Git. Following best practices for ontology publishing, *STO* is available via a *W3ID* permanent URL;[1] the ontology is also registered in the Linked Open Vocabulary service,[2] as well as in the OntoPortal, a resource for publishing industrial ontologies.[3] Additionally, *STO* is published under a Creative Commons license. A summary of the characteristics of the *STO* ontology is reported in Table 4.1.

Ontology Overview

In the following, core classes of *STO* are described (cf. Figure 4.4).
Standard: represents a standard; since standards are defined as documents, it specializes the `foaf:Document` class to model standards.
sto:StandardOrganization: describes organizations that develop standards such as *ISO*, *IEC*. This class specializes the `foaf:Organization` class.

[1] https://w3id.org/i40/sto
[2] http://lov.okfn.org
[3] http://iofportal.ncor.buffalo.edu/ontologies/STO

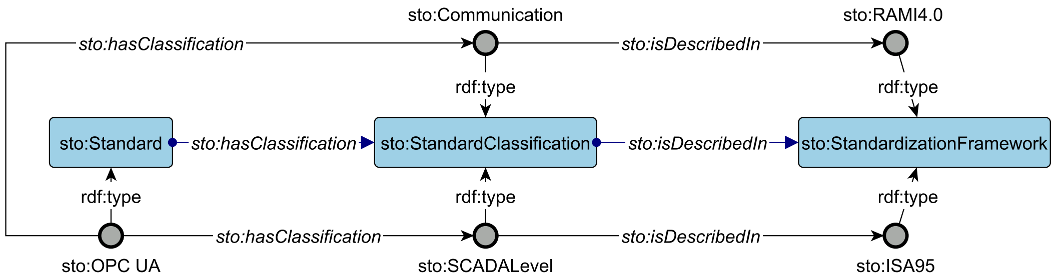


Figure 4.5: **Example of the classification of the OPC UA standard in *STO***. OPC UA is classified with two instances of the `sto:StandardClassification` class, i.e., `sto:SCADALevel` and `sto:Communication`. These instances are connected to their standardization frameworks, i.e., `sto:ISA95` and `sto:RAMI4.0`, respectively.

sto:StandardClassification: this class models classifications of standards that are described by different standardization frameworks. The classes `rami:ITDimension`, `rami:HierarchyLevel`, and `rami:AdminShellSubmodel` are external to *STO*. These classes are used to model RAMI4.0 dimensions and classification of standards according to the Administration Shell concept. Internal classes, e.g., `sto:ProductionSystemLifeCycle`, `sto:ProductDevelopmentLifeCycle`, and `sto:ISA95Level` describe the classification of standards provided by the NIST standardization landscape. These classes are instantiated to express the values of standard classifications, e.g., `sto:ISA95Level` comprises instances as `sto:SCADALevel` (cf. Figure 4.5). Standards are then connected through these instances to represent their classification, e.g., OPC UA is connected to the framework level in IICF as well as to the communication level in RAMI4.0. Finally, classifications of standards are linked to standardization frameworks in which they are categorized, e.g., framework level is described in IICF.
sto:Domain: specifies relevant domains for standards, e.g., Manufacturing Operation Management, Functional Safety, and Machine to Machine Communication.
lexvo:Language models the language in which the standards are available, e.g., English, German.

Description of Properties

The core properties of *STO* are described in this section.
sto:hasPublisher: connects a standard with the organization that published it. Similarly, **sto:hasDeveloper** links the standard to the organization that developed it.
sto:hasOfficialResource: points to the official (s) websites describing the standard.
sto:hasTag: In some cases, to refer to well-known standards the tag of the standard is utilized, i.e., OPC UA. The property `sto:hasTag` is used to represent this relationship.
dcterms:license: An external property that links a standard with its correspondent license document.
sto:hasDBpediaResource: allows to interlink *STO* instances with DBpedia. This property is considered as a subproperty of `owl:sameAs`.
dcterms:hasVersion: describes the edition or version of a given standard.
sto:relatedTo: represents links between I40 standards. This property is defined as symmetric and transitive. The inference model based on `sto:relatedTo` allows for uncovering new relations

Table 4.1: **Summary of the *STO* Ontology Characteristics.** The table shows a summary of the *STO* ontology in aspects such as general details, reused ontologies, documentation, naming conventions, multilinguality, and availability.

General	**Name**	Standard Ontology (sto)
	Size	53 classes, 30 object properties, 24 data properties, 700 individuals
	DL Expressivity	SHOIF(D)
Reuse	**Reused Ontologies**	DCTERMS, PROV, DUL, FOAF, RAMI4.0, OM
	Reused ODPs	Componency ODP
Documentation	**All elements documented**	By means of `rdfs:label`, `rdfs:comment`, `skos:prefLabel` and `rdfs:isDefinedBy`
Naming Conventions	**For schema and individuals**	CamelCase notation for the schema and Ada for individuals
Multilinguality	**English labels for all terms**	`rdfs:label` and `rdfs:comment` with the @en notation
Availability	**PersistentURI**	`https://w3id.org/i40/sto`
	GitHub	`https://github.com/i40-Tools/StandardsOntology`
	LOV	`http://lov.okfn.org/dataset/lov/vocabs/sto`
	OntoPortal	`http://iofportal.ncor.buffalo.edu/ontologies/STO`
	Licence	Creative Commons 3.0
	VoCol Instance	`http://vocol.iais.fraunhofer.de/sto/`

between standards. For example, the standard *ISO 13849* is connected to the standard *IEC 61511* by using the `sto:relatedTo` property. Likewise, *IEC 61511* is connected to *IEC 61508*. By utilizing the transitivity encoded in this property, a relation between the standards *ISO 13849* and *IEC 61508* can be inferred.

prov:wasGeneratedBy: specializes the `sto:relatedTo` property; it represents the relation in which one standard is derived from other standard, e.g., *DIN SPEC 16592* is derived from the OPC UA and the AML standard.

dul:isComponentOf: describes relations of standards which constitute components or parts of other standards. For example, the CAEX standard, i.e., *IEC 62424* is part of the AML standard. This property is also a subproperty of `sto:relatedTo`.

sto:hasEdition: indicates the current edition of the standard.

sto:hasStabilityYear: defines the year that the stability period is finished and the current version of the standard is valid.

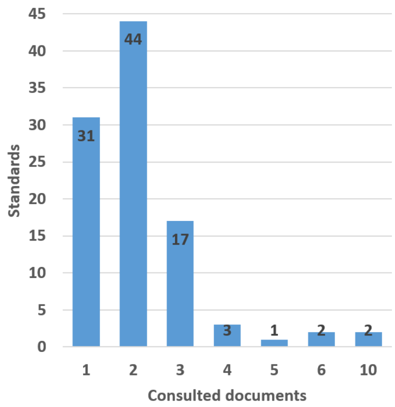
sto:hasOntology: refers to the ontology of a standard, in case it has been already defined. For instance, the *ISO 15926* standard, used in the integration of life-cycle data for process plants, is available as an ontology.[4]

4.1.2 Extract Information of Standards

To extract information about standards, we searched documents of standardization frameworks for I40. Figure 4.6(b) shows the utilized criteria for searching relevant documents related to standardization frameworks for I40. We started by combining terms such as “Industry 4.0”, “Reference Architectures”, “Standardization Landscape”, and “Standards”. These combinations allowed to retrieve documents of standardization frameworks that categorize standards, e.g., the RAMI4.0 Model [20] and the IIRA architecture [7]. By following this search process a list of documents of standardization frameworks describing standards is compiled. For each framework in the list, an RDF molecule with the name and URI of the framework are created in the I40KG.

Furthermore, to retrieve organizations that publish or develop standards, we used the terms “Standardization Organizations”, “Industry 4.0”, “Standards”, and “Reference Architectures”.

[4] https://www.posccaesar.org/wiki/ISO15926inOWL



(a) **Documents per standards**

Search engines	Keywords	Period
google.com, google schoolar, ask.com, bing.com	“Standardization Frameworks”, “Standards”, “Relations”, “Smart Manufacturing”, “Industry 4.0”, “Industrie 4.0”, “Reference Architectures”, “Standardization Landscape”, “Internet of Things”, “Standardization Organizations”	12.2002 - 07.2017

(b) **Search criteria for selecting documents**

Figure 4.6: **Number of consulted documents per standards and utilized search criteria.** Figure 4.6(a) shows the number of consulted documents in the X-axis. The Y-axis depicts the number of standards molecules that are created with a given number of documents, e.g., to create the molecules of AML and OPC UA, ten documents are consulted. Figure 4.6(b) outlines the search engines, the keywords as well as the period in which the retrieved documents are published.

The terms are introduced on the Web engines and the first top ten documents are retrieved and analyzed. We compiled a list of 30 standardization organizations. For each organization in the list, an RDF molecule with the name of the standardization organization and its URI is introduced in the I40KG. An RDF molecule is defined as a set of triples sharing the same subject [61]. The name of the organization is searched in the Web engines. In case the organization contains an official page, its link is added to the RDF molecule. From its official page, the acronym and the formation date are retrieved. Additionally, if the links to Wikipedia and DBpedia exist, they are also added to its molecule. Listing 4.1 depicts the RDF molecule of the *OPC Foundation* as instance of the class `sto:StandardOrganization`. OPC Foundation is the organization that developed the OPC UA standard.

```
@prefix sto: <https://w3id.org/i40/sto#> .
@prefix rdfs:  <http://www.w3.org/2000/01/rdf-schema#> .
sto:OPC_Foundation           a  sto:StandardOrganization ;
    rdfs:label               "OPC Foundation"@en;
    rdfs:comment             "Industry consortium that creates and maintains..."@en;
    sto:abbreviation         "OPC";
    sto:formationDate        "1994-01-01"^^xsd:date;
    sto:hasDBpediaResource   <http://dbpedia.org/page/OPC_Foundation>;
    sto:hasOfficialResource  <https://opcfoundation.org/>;
    sto:hasWikipediaArticle  <https://en.wikipedia.org/wiki/OPC_Foundation>.
```

Listing 4.1: **Description of the RDF Molecule of the OPC Foundation**

Then, Algorithm 1 computes a list of standards which are described in the retrieved documents of the standardization frameworks. Standards are searched on the retrieved documents using the pattern “publishing organization” and “numeric value”, e.g., IEC as the organization and 62541 as the numeric value. In addition, the algorithm creates a list with the mappings among the standards and the standardization frameworks.

Algorithm 1 Extract information of standards from documents

Input: stdFrameworkDocList, stdOrganizationList
Output: StandardList, MappingsStandardFrameworkList

procedure ExtractStandardsInfo
  for each doc in $stdFrameworkDocList$ **do** :
    $stdFramework = stdFrameworkDocList.getStdFramework()$
    $std = doc.searchStandardByName([PublishingOrganization + numericValue])$
    $createStdList(\langle std, URI\rangle)$
    $createMappingList(\langle std, stdFramework\rangle)$

The mapping to standardization frameworks enables the link of existing standards to respective frameworks. We investigated standardization frameworks comprising information about classifications of standards. Typically, this classification is performed in dimensions or layers and is made according to the function of the standard. RAMI4.0 classifies standards according to three general dimensions, i.e., *IT*, Life-Cycle and Value Stream, and the Hierarchy Level. For example, the *IT* dimension of RAMI4.0 is analyzed. The layers which belong to the *IT* dimension are considered, from the *Asset* to the *Business* layer (cf. Figure 4.1(a)). Furthermore, standards are mapped to the specific layers of this dimension. The Administration Shell can be considered as other standardization framework that classifies standards. In this case, it delineates how standards are linked to submodels. The submodels enclose the different functions of that an asset requires, e.g., identification, communication, or engineering [6]. The Identification submodel is aligned with the *ISO 29005* standard, whereas the Communication submodel with the *IEC 61784 Fieldbus Profiles* (cf. Figure 2.2). For instance, the engineering submodel is aligned with standards such as *IEC 61360*, *IEC 61987*, and *eCl@ss*.

The step of extracting information of standards from unstructured data sources allows for representing this knowledge using the *STO* ontology. Furthermore, the knowledge is encoded in the I40KG. Hence, the interoperability conflicts **SIC1**, **SIC2**, and **SIC4** existing across the data sources describing standardization frameworks and standards, are resolved during this step.

4.1.3 Knowledge Graph Population

The population of I40KG is performed according to the *STO* ontology. To populate the I40KG, we rely on the concepts of RDF molecules and RDF molecule templates (RDF-MTs). An RDF-MT is an abstract representation of the set of properties associated with an RDF class, and the links between the class with other RDF classes [164]. Instances of an RDF-MT correspond to RDF molecules in a knowledge graph. The RDF-MTs describe the relations between classes in a knowledge graph and the classes of the knowledge graphs to which it is linked. Algorithm 2 details the creation of RDF molecules for standards. The input to the algorithm is a list comprising the labels of standards retrieved in Algorithm 1. Next, the properties of the standard RDF-MT are given as input. Finally, a list containing the mappings among standards and standardization frameworks is also an input to the algorithm.

To prevent the duplication in the creation of the same RDF molecule, a unique URI is defined for each name of standard. Then, the algorithm iterates over the list of standards and searches on the web engines. Based on this search, the top 30 documents are selected. A total number of 220 documents of different types are retrieved, i.e., technical reports (12), white papers (6), scientific articles (28), standard specifications (165), technical presentations (7), and technical

Algorithm 2 Standards RDF Molecule creation

```
Input: StandardList, StandardMT, MappingsStandardFrameworkList
Output: StandardRDFMoleculeList
procedure CreateStandardRDFMolecule
    for each ⟨std, stdLabel⟩ in StandardList do :
        standardRDFMolecule.createURI(stdLabel)
        listOfStdDocuments = searchOnWebEngines(stdLabel)
        listOfStdDocuments = listOfStdDocuments.getTop30()
    for each doc in listOfStdDocuments do :
        stdProperty = doc.searchStdProperty⟨stdLabel, StandardMT⟩
        if exists(stdProperty) then:
            standardRDFMolecule.addPropertyValue(stdProperty, valueProperty)
            StandardRDFMoleculeList.add(standardRDFMolecule)
        if doc.isWikiPedia then:
            standardRDFMolecule.addWikiPediaLink(doc)
            standardRDFMolecule.addDBPediaLink()
    for each ⟨std, stdFramework⟩ in MappingsStandardFrameworkList do :
        standardRDFMolecule.addMapping(stdFramework)
        StandardRDFMoleculeList.add(standardRDFMolecule)
```

papers (2). The period in which these documents were published ranges from December 2002 to July 2017. Figure 4.6(a) depicts the number of consulted documents required to create RDF molecules for each standard. For each one of the documents, the properties of the standard RDF-MT are examined. The obtained values are used to build one RDF molecule for each standard. Furthermore, the mappings to the standardization frameworks are added to the standard RDF molecule. The linking to external knowledge graphs is a common method for

Algorithm 3 Create relations between standards

```
Input: StandardRDFMoleculeList, StandardMT
Output: Relations created on the molecules of standards
procedure CreateRelationsStandards
    for each ⟨std_i, std_j⟩ in StandardRDFMoleculeList do :
        for each property in StandardMT do :
            if std_i ≠ std_j then:
                relation = searchOnWebEngines(std_i, property, std_j)
                if exist(StandardMT.relation) then:
                    addRelation(std_i, relation, std_j)
```

knowledge graph completion [43]. Furthermore, the I40KG is linked to *DBpedia* [41]. To perform this link, the name of the standards is inspected. In case that the name exists in Wikipedia, e.g., https://en.wikipedia.org/wiki/IEC_61131, it is also present in DBpedia with the same name, e.g., http://dbpedia.org/page/IEC_61131. Then, the property `sto:hasDBpediaResource` is employed to connect the standard to the link in DBpedia (cf. Listing 4.1). The output of the algorithm is a list of the standard RDF molecules comprising also the mappings among standards and standardization frameworks.

Table 4.2: **Basic and optional properties for the RDF molecule of the OPC UA standard.** Values for the properties of the OPC UA standard are extracted from the consulted documents. Basic properties required to be always considered when creating a new RDF molecule for standards whereas optional properties are only considered if they are found in the documents.

Property	Description	Example	Type
Name	Official name	OPC UA	Basic
Tag	Tag used to refer to the Standard	OPC UA	Optional
Description	Description of the function	International standard for vertical and horizontal communication in manufacturing and automation, providing semantic interoperability for the world of connected systems.	Basic
Publisher	Organization (s) responsible for publishing	OPC Foundation	Basic
Developer	Organization (s) responsible for developing	OPC Foundation	Basic
Official resource	Official Website describing information	https://opcfoundation.org/about/opc-technologies/opc-ua/	Basic
Publication date	The publication date is required since standards are reviewed every five years	10.01.2008	Basic
Domain	Domain of application or use	Machine to Machine (M2M) Communication	Optional
License	License under which the standard is published	GPLv2	Optional
Edition	Last edition of the Standard document	1.02	Optional
Relation	Relations with other standards reported by the literature	OPC UA – AML, “interoperability” [9], “integration” [165]	Optional
Classification	Classification of the standards w.r.t. standardization frameworks layers and levels	OPC UA and the communication layer of the RAMI4.0 model (cf. Figure 4.5)	Optional
DBpedia	RDF representation of the Standard	http://dbpedia.org/page/OPC_Unified_Architecture	Optional

Further, to create relations between standards, the Algorithm 3 searches for connections between pairs of standards. These connections are compared with the properties of the standard RDF-MT. The property `sto:relatedTo` describes that two standards are mentioned in one document but no explicit relation is defined. In case that explicit relations are defined, their name is encoded as properties. For instance, relations such as “interoperability” [9] and “integration” [165] are encoded with the properties `sto:isInteroperableWith` and `sto:integratesWith`, respectively. These properties represent explicit references to the relation between two standards and are modeled as subproperties of `sto:relatedTo`.

Table 4.2 summarizes the values of the main properties describing the OPC UA standard whereas Listing 4.2 illustrates them in Turtle format. The I40KG comprises more than 200 standards and more than 25 standard organizations. Moreover, 103 direct relations between standards are encoded as a part of the knowledge graph. I40KG is openly available and can be expanded by the community with interest in I40 as well as domain experts in this topic by directly accessing it on Github.[5] The I40KG comprises the description of standards, along with their metadata and inter-relations. Additionally, it contains information regarding classifications of standards according to different standardization frameworks. Furthermore, the descriptions of the organizations that published the standards are included as well. The RDF-MT for standards and standardization frameworks unifies the way in which the representation of these entities is made, **SIC1** and **SIC2** are resolved. In addition, the mapping of standards with standardization frameworks enables the solution of the **SIC1** conflict among them.

[5] https://github.com/i40-Tools/StandardsOntology

```
@prefix sto:  <https://w3id.org/i40/sto#> .
@prefix rami: <https://w3id.org/i40/rami#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

sto:OPC_UA                      a sto:Standard;
    rdfs:label                  "OPC UA"@en;
    rdfs:comment                "International standard for vertical and horizontal communication..."@en;
    sto:hasTag                  "OPC UA"@en;
    sto:hasPublisher            sto:OPC_Foundation;
    sto:hasDeveloper            sto:OPC_Foundation;
    sto:hasDBpediaResource      <http://dbpedia.org/resource/OPC_Unified_Architecture>;
    sto:hasOfficialResource     <https://opcfoundation.org/about/opc-technologies/opc-ua/>;
    sto:hasWikipediaArticle     <https://en.wikipedia.org/wiki/OPC_Unified_Architecture>;
    sto:isInteroperableWith     sto:AML;
    sto:integratesWith          sto:IEC_61499;
    sto:hasDomain               sto:M2MCommunication;
    sto:hasClassification       rami:Communication, sto:FrameworkLevel ;
    dcterms:license             sto:GPLv2.
```

Listing 4.2: **Description of the RDF molecule of the OPC UA standard**

4.1.4 Knowledge Graph Integration

The knowledge integration step permits to semantically define connections between instances in I40KG to resolve semantic interoperability conflicts. For example, there are cases when same standards have the same meaning but are named differently, i.e., a semantic interoperability conflict **SIC1** exist between those standards. This applies to the *IEC 62541* standard, which is actually the OPC UA standard published by the IEC organization but known with a different name in the IEC publication. In this case, an additional set of properties for the RDF standard molecule of IEC 62541 are of interest, e.g., available languages, the technical committee, as well as the stability date. These properties are extracted from the official website of the IEC[6] which is retrieved in the Algorithm 1.

[6] https://webstore.iec.ch/publication/21996

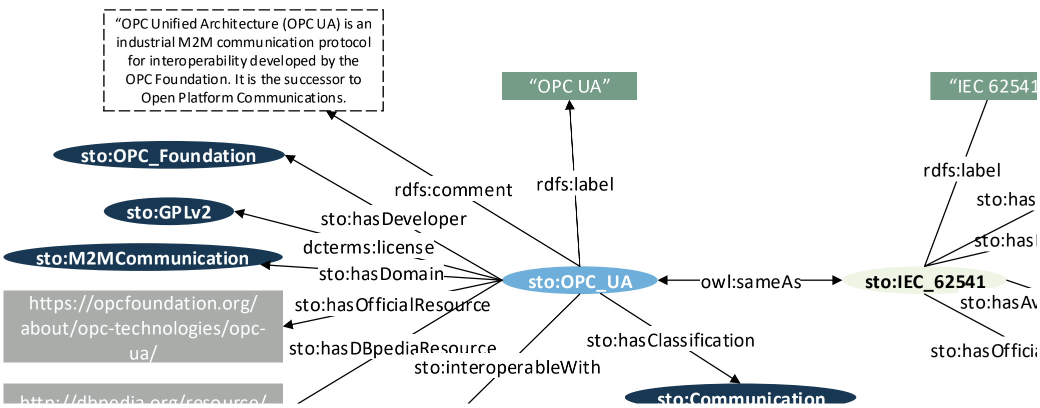
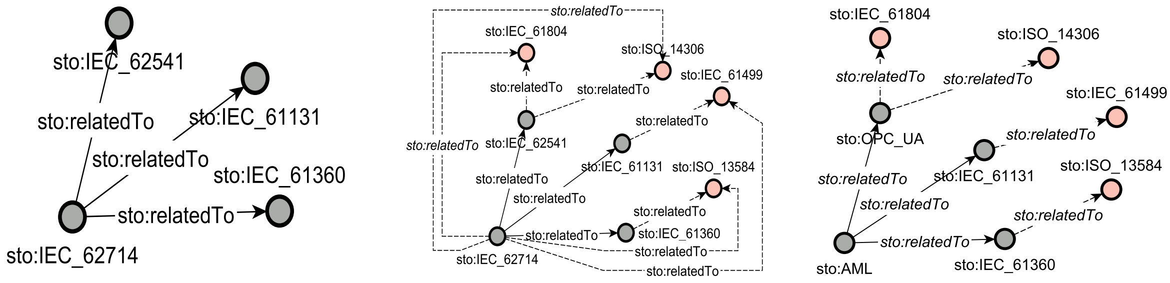


Figure 4.7: **Example of the integration of the RDF molecules of the standards OPC UA and IEC 62541**. The left side of the figure shows the RDF molecule of the OPC UA standard whereas the right side depicts the RDF molecule of the IEC 62541 standard. The predicate `owl:sameAs` is used to semantically link the two RDF molecules and resolve the conflict **SIC1**.



(a) Explicit relations between AML and other I40 standards

(b) Discovered relations between AML and other I40 standards

(c) Explicit and inferred relations between AML and I40 standards

Figure 4.8: **I40 Standards related to AML**. Relations between I40 standards are visualized using graphs; continuous and dashed directed arrows represent explicit and inferred relations, respectively. The inference model relies on the transitive and symmetric properties of `sto:relatedTo`. (a) Known relations between AML and I40 standards are described using the property `sto:relatedTo`. (b) Relations between I40 standards connected to AML with dashed directed arrows and colored in a different color, are inferred. The relation between AML and the standard of Measurement and Control Devices (IEC 61499) has been validated in the literature [166].

These standards are named differently depending on the organizations but they refer to the same standard, thus they are considered as the same entities and integrated into the knowledge graph. In this case, we used the **Union** policy to create an RDF molecule which combines the knowledge of the two RDF molecules of the OPC UA and IEC 62541 standards.

4.1.5 Knowledge Graph Reasoning

One of the main motivations to create I40KG is to encode the knowledge of I40 standards as well as to study existing relations among them. The internal reasoning step is performed with the aim to unveil new knowledge in the I40KG.

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX sto: <https://w3id.org/i40/sto#>

SELECT  ?firstStd  ?p  ?secondStd
WHERE {
{
    ?s  sto:relatedTo+|^sto:relatedTo  ?o .
    BIND  (STR(sto:relatedTo)  as  ?p) .
}
UNION {
    ?firstStd  ?p  ?secondStd .
    ?p  rdfs:subPropertyOf  sto:relatedTo .
} }
```

Listing 4.3: **Searching for AML and related standards**

Figure 4.8(a) depicts explicit relations which are currently annotated in the knowledge graph; Figure 4.8(c) shows inferred relations which are obtained after executing the query described in Listing 4.3 and running the inference process based on the symmetric and transitivity properties of `sto:relatedTo`. These queries can be evaluated on the *STO* VoCol repository.[7]

[7] http://vocol.iais.fraunhofer.de/sto/

Metrics	**I40KG no reasoning**	**I40KG reasoning**
Num of nodes	68	68
Num of edges	66	227
Graph density	0.025	0.068
Avg. num of neighbors	1.706	4.559
Connected components	16	16
Transitivity	0.12	0.732
Clustering coefficient	0.085	0.389
Graph centralization	0.128	0.237

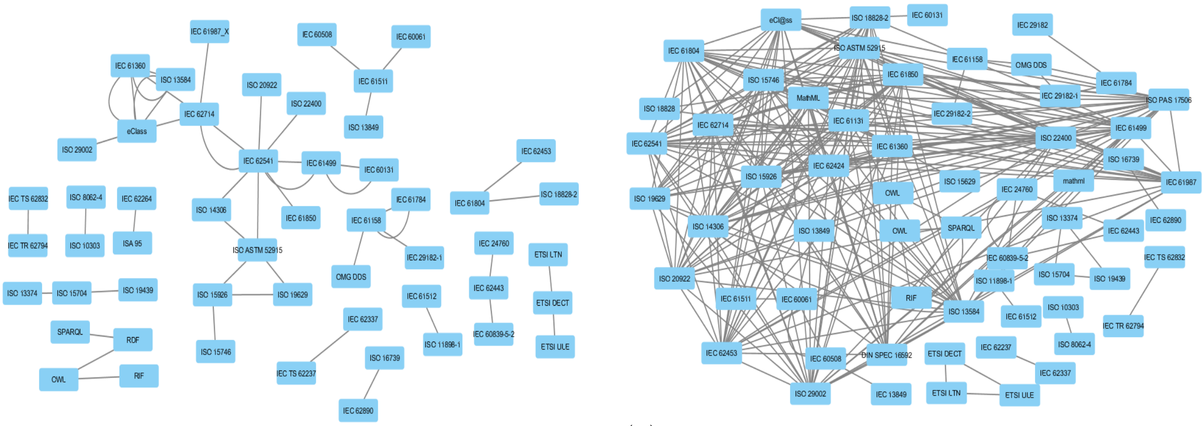
Table 4.3: **Graph metrics for the I40KG before and after the internal reasoning step.** Some metrics are considered for evaluating the I40KG before and after the internal reasoning step, i.e., from the number of nodes to the graph centralization. Results reveals a general improvement in the structure of the KG, e.g., Clustering coefficient (0.085 to 0.389) suggests a tendency of the standards who share the same connections to become connected.

This step allows for retrieving explicitly-defined relations in I40KG as well as those that are inferred. For instance, relations of AML and OPC UA, OPC UA and IEC 61499 are explicitly defined in the KG. Based on this fact, the relation between AML and IEC 61499 (cf. Figure 4.8(c)) is inferred. Moreover, the existence of this relation is checked and validated in the literature and is of importance for the domain [166]. Furthermore, relations between standards in the I40KG can be retrieved. The result of this query retrieves 103 relations between the standards (cf. Figure 4.9(b)), and 266 new relations inferred when the symmetric and transitive properties are considered by the inference process (cf. Figure 4.9(a)).

Several graph metrics are computed over the two graphs, i.e., with and without reasoning to analyze the connectivity and relationships discovered during the reasoning step. Table 4.3 reports on these metrics. As observed, the number of edges increases from 66 to 227, indicating that new relations between standards are discovered. The graph density–fraction of the number of potential connections in the graph that are actual connections– increases slightly; it goes from 0.025 to 0.068. This implies a slight improvement of the connections among the standards. Values of transitivity are augmented, i.e., from 0.12 to 0.732. This indicates an increment of the possibility that a relation between two standards in the graph is transitive. The clustering coefficient also increases from 0.085 to 0.389. These results highlights the increment of the degree to which the standards, who share the same connections, tend to cluster together. The graph also becomes more centralized, i.e., from 0.128 to 0.237. These findings reveal the importance of standards within the I40KG. For instance, OPC UA with a value of centrality of 0.8, seems to be more important than ISO 20922 with a value of 0.58. By enabling the discovery of new relations among standards, this steps resolve the conflicts **SC2**, **SC3**, and **SC4**.

4.1.6 Knowledge Graph Interlinking

The knowledge graph interlinking step enables interconnecting I40KG with knowledge graphs in the Linked Open Data Cloud (LOD) [40], i.e., DBpedia. Algorithm 4 describes the followed procedure. Initially, every RDF molecule in the I40KG is surveyed. Then, whenever the RDF molecule comprises a link to DBpedia, the correspondent knowledge its extracted and added to the I40KG. This knowledge is expressed in form of new classes, properties, and instances, which are added to the I40KG in the interlinking step. DBpedia properties that do not add semantic



(a) Explicit relations of I40 standards

(b) Explicit and inferred relations between I40 standards

Figure 4.9: **Relations between I40 Standards**. Relations between I40 standards are visualized using graphs. The inference model relies on the transitive and symmetric properties of `sto:relatedTo`. (a) Known relations between I40 standards are explicitly described using the property `sto:relatedTo`. (b) Relations between I40 standards are inferred; the graph comprises 316 edges: 266 are inferred while 66 are explicit.

Algorithm 4 Interlinking I40KG with DBpedia

Input: I40KG, DBpediaSPARQLEndpoint
Output: interlinked I40KG

procedure INTERLINKI40KG
    for each $RDFMolecule$ in $I40KG$ **do**:
        if $isLinkedToDBpedia(RDFMolecule.predicate)$ **then**:
            $results = DBpediaSPARQLEndpoint.query(RDFMolecule.subject)$
            for each $results$ **do**:
                $RDFMolecule.add(result.predicate, result.object)$

value to I40KG, e.g., `dbo:wikiPageID`, `dbo:wikiPageRevisionID`[8] are omitted. The knowledge graph interlinking is able to discover new knowledge of standards and standard organizations. The discovered knowledge enhance the RDF-MT, i.e., the schema definition of the involved entities. This permit to resolve the conflicts **SC2** and **SC4**.

4.2 Evaluation

The quality and accuracy of the semantic integration techniques proposed in this article are empirically studied by relying on a retrospective evaluation. The retrospective evaluation involves humans to label errors of the results [43]. Typically, in this type of evaluation precision is the utilized quality metric. The retrospective evaluation is performed and particularly, the following research questions are investigated:

- Can a *knowledge graph driven* approach allow for the discovery of valid relations among standards?;

[8] We use the dbo prefix as the namespace of the http://dbpedia.org/ontology/

Table 4.4: **Evaluation of the discovered relations in the knowledge graph interlinking step.** I40KG comprises initially 66 explicit relations. A total of 266 new relations are inferred and 188 are validated by searching into different types of documents; the method reaches a 0.71 value of precision.

Initial Relations	Validated	Total	Precision
66	188	266	0.71

- Is the proposed *knowledge graph interlinking step* able to discover new knowledge in terms of classes, properties, and instances?

In the following, we present the experiments performed to evaluate our research questions.

4.2.1 Discovering Relations between Standards

With the goal of investigating the effectiveness of semantic integration approach, discovered relations between standards are checked. In order to perform the search to the 266 discovered relations, a combination of the names and tags of the standards is used, e.g., "AML" and "IEC 61499". As a result, different types of documents are retrieved, i.e., scientific articles (66), white papers (8), standard specifications (17), technical reports (48), technical presentations (6), thesis (3), for a total of 148 documents. These documents are different than those employed to create I40KG. The target is to validate that the discovered relations between standards existed in the consulted documents. Two criteria are utilized to determine whether the discovered relation can be evaluated as true: 1) direct relations between standards; and 2) in case standards appeared in the same document indicating the same or similar goal, e.g., I40, Smart Manufacturing. General catalogs that list the name of standards are not considered. To find those documents, different search engines are used, such as Google, Google Scholar, Bing, Yahoo and Ask. Table 4.4 reports on the results of the study. A total of 188 relations are positively validated out of 266 new relations inferred for a precision of 0.71. Thus, these results allow to positively answer **RQ1**.

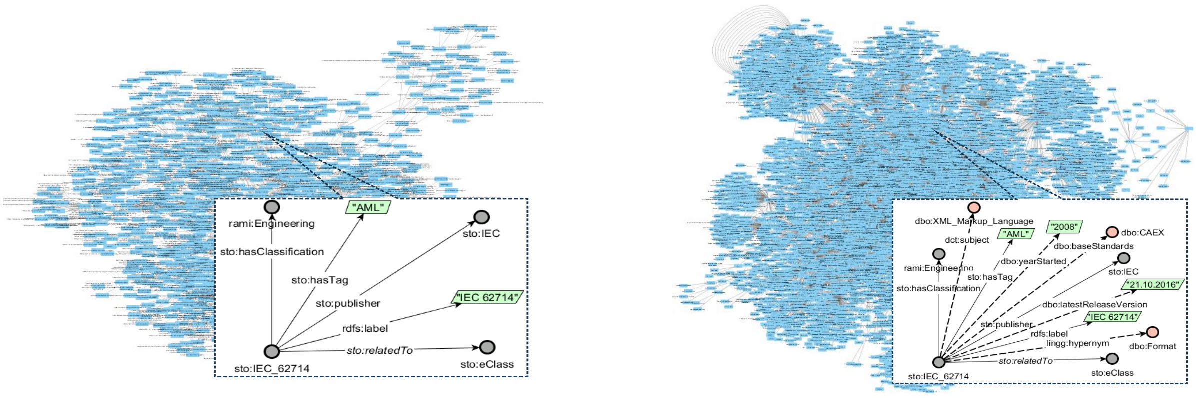
4.2.2 Discovering Knowledge through Knowledge Graph Interlinking

With the objective to measure the effectiveness of the knowledge graph approach after the linking step, four criteria are studied. These criteria are computed by considering instances of standards and standard organizations.

Table 4.5: **Precision values for properties related to the RDF-MTs of standards and standard organizations after the knowledge graph interlinking step**. Precision values for new class linkings, i.e., `rdf:type` and the total number of new properties after executing the knowledge graph interlinking step are reported. Instances of standards (**Std**) and standard organizations (**Org**) are examined.

Criteria	Total Std	Total Org	Precision Std	Precision Org
New class linkings	93	108	0.66	0.90
New properties	35	39	0.97	0.96

- *Number of new class linkings*. This refers to the number of new classes that are automatically added to the instance as types. For example, the DBpedia class *Industry XML Specific Standards* is added to the AML standard as one of its types. This adds meaning to the instance of the standard since AML is an industry standard and is also XML based.



(a) I40KG

(b) I40KG after interlinking step

Figure 4.10: **I40KG before and after the knowledge graph interlinking step**. New knowledge is added to I40KG after the knowledge interlinking step. The number of triples increase from 6336 to 9336. An instance of the standard RDF-MT is depicted before and after the step, i.e., the AML standard. The image depicts part of the added knowledge to the standard RDF-MT, i.e., the property `dbo:yearStarted` with the literal “2008”, stating the year in which the standard started. This is an important property for the standard RDF-MT which is added to I40KG. A total of 35 new properties are added for the standard RDF-MT and 88 are added for the RDF-MT of standard organizations with a precision of up to 96%.

In this case, the repetitions of classes are not considered, i.e., if one class is computed for one standard then it is not computed again.

- *Number of new properties.* This criteria refers to the number of new properties that are automatically added to each instance of the standards. For instance, the AML standard is enriched (among others) with the properties `dcterms:subject`, and `dbo:yearStarted` which are not considered in I40KG (cf. Figure 4.11). In this case, two properties are assessed for the number of new properties. A manual inspection is performed over the obtained properties. In case the property is not defined in DBpedia as an `rdf:property` is counted as a false positive. Further, when the property does not add value nor have a description is also not considered.

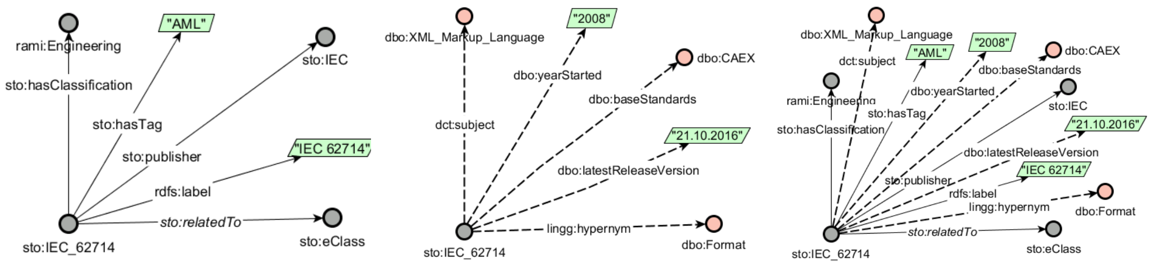
To materialize the results of the defined criteria, one SPARQL query for each one is defined. Next, queries are executed on top of the enriched I40KG. The enriched KG[9] and the queries[10] are publicly available in github. Table 4.6 reports on the applied queries to the number of standard (73) and standard organization instances (22) which contain a link to DBpedia. One can observe from the table that new knowledge for standards and standard organizations is discovered, i.e., new classes, properties and triples. As observed, the precision increases by up to 0.96, suggesting thus that the accuracy of the discovered relations is increased by up to 96%.

4.2.3 Effectiveness of the Knowledge Graph Interlinking Step

The knowledge graph interlinking step is evaluated in order to measure the quality of the generated connections. The automatic linking of RDF knowledge graphs can be prone to

[9] https://github.com/i40-Tools/StandardOntology/blob/master/sto_enriched.ttl
[10] https://github.com/i40-Tools/StandardOntology/tree/master/Queries/Criteria

different types of errors, i.e., syntactic, logical, and semantic errors. After the knowledge graph interlinking step, the soundness of the discovered relations is validated. Relation validity is affected by syntactic and semantic errors in the linked knowledge graphs [167]. For example, the RDF triple `sto:SCOR rdf:type dbo:Person` in DBpedia indicates that the standard SCOR is a person. Links from I40KG to the resource `sto:SCOR` in DBpedia also introduce this error in I40KG and affect the quality of the represented relations. During the evaluation of the effectiveness of the interlinking process, the validity of the relations is measured in terms of precision.



(a) Initial AML molecule (b) New connections (c) Final molecule enrichment

Figure 4.11: **Example of the RDF molecule for the AML standard before and after the knowledge graph interlinking step**. Figure 4.11(a) shows the basic RDF molecule of one standard in the I40KG, i.e., `sto:AML` before the knowledge graph interlinking step. Figure 4.11(b) depicts five of the new connections, properties, instances, and literal values that are incorporated for the `sto:IEC_62714` standard. Figure 4.11(c) illustrates the complete molecule for `sto:IEC_62714` after the knowledge graph interlinking step.

Table 4.6: **Precision values for studied properties of the RDF-MT of standard and standard organizations after the knowledge graph interlinking step**. Values of three properties are observed, i.e., `owl:sameAs` links, `dcterms:subject`, and `lingg:hypernym`. **Precision** is computed based on the values of these properties for instances of standards (**Std**) and standardization organizations (**Org**).

Criteria	Total Std	Total Org	Precision Std	Precision Org
`owl:sameAs`	301	217	0.98	0.91
`dcterms:subject`	144	115	0.98	0.96
`lingg:hypernym`	44	36	0.61	0.85

These properties are related to the values of the instances they are linked to. Second, the number of new class linkings encountered by means of the the `rdf:type` property are studied. These class linkings add meaning to the standard or organization instance they are linked. For example, in case of the following is encountered `sto:SCOR rdf:type dbo:Person`. This is considered as a semantic error and the value is marked as false, since the `sto:SCOR` is a standard and not a person. Similarly, in case that SCOR is classified as a model, i.e., `sto:SCOR rdf:type yago:Model`[11] is considered as true. SCOR, is the acronym of *Supply Chain Operation Reference Model*. Consequently, the existing classification in DBpedia is correct and considered as a true positive for the interlinking step. Values of the `dcterms:subject`, are also inspected regarding semantic errors. By definition, this property describes the topic of a given resource.[12] Based on

[11] We use yago as prefix of the YAGO knowledge graph: http://yago-knowledge.org/resource/
[12] http://dublincore.org/usage/terms/history/#subjectT-001

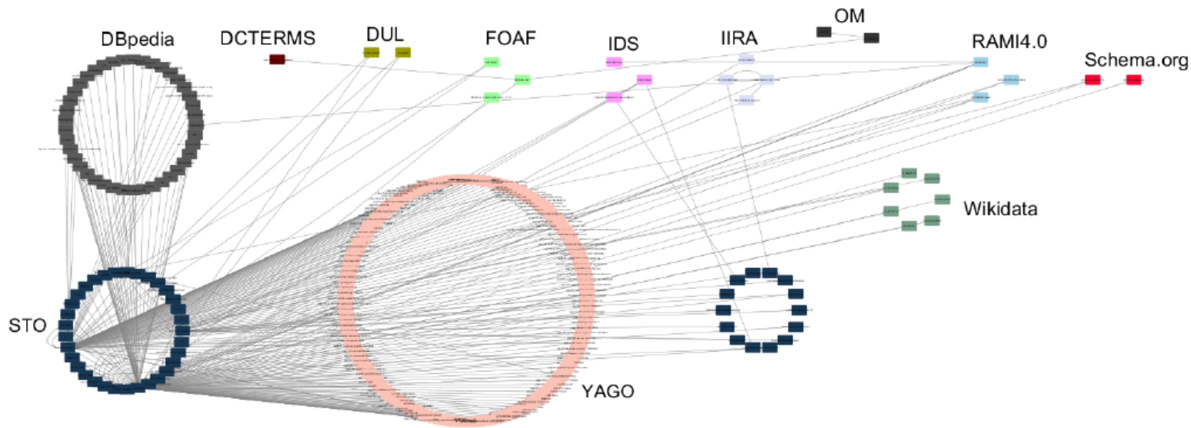


Figure 4.12: **Analysis of I40KG after the knowledge graph interlinking step**. The graph comprises 221 RDF-MTs and 259 intra- and inter-knowledge graph links. Used knowledge graph are represented in circles by different colors. The line between dots in the same circle shows intra-knowledge graph links, while a line between dots in different circles corresponds to inter-knowledge graph links. YAGO and DBpedia are the most utilized knowledge graphs. As expected, two RDF-MTs, i.e., `sto:Standard` and `sto:StandardOrganization` are the source of most of the generated links.

this definition, values are observed by checking whether they correctly represent a topic or not. A similar process is conducted for the property `lingg:hypernym`. We checked whether the linked resource is a hypernym, i.e., a broader classification of the instance. For instance, the standard *B2MML*, which is defined as an XML implementation of the ANSI/ISA-95 family of standards, has *implementation* as its hypernym. In this case, the new link is correct and we counted it as a true positive; otherwise as a false positive.

With respect to the `owl:sameAs`, links to other instances representing the same semantics are revised. The `owl:sameAs` links pointing to non existing resources like *rdf.freebase.com* are not considered. Most of the `owl:sameAs` links are with different versions of DBpedia in other languages, e.g., German, French, Spanish, Italian, Japanese. Additionally, important knowledge graphs such as *Wikidata* and *YAGO* are also interlinked. The values of the four properties are inspected for each one of the instances enriched during the knowledge graph interlinking step, i.e., 103 instances of standards, and 23 of standard organizations. Table 4.5 reports on the computed values of precision.

4.2.4 RDF Molecule Templates

Table 4.6 shows the computed precision for the values of the properties added to the RDF-MT of standard (35) and standard organization (86) after the knowledge graph interlinking step, respectively. Precision values for studied properties, i.e., `owl:sameAs`, `dcterms:subject`, and `lingg:hypernym` of the RDF-MT of standard and standard organizations after the knowledge graph interlinking step. Precision is computed based on the values of these properties for instances of standards and standardization organizations. Relations for the RDF-MTs based on these properties can be validated by up to 98%.

Metrics	I40KG	I40KG interlinked
Num of nodes	86	256
Num of edges	108	259
Graph density	0.025	0.009
Avg. num of neighbors	2.14	2.23
Connected components	1	2
Transitivity	0.017	0.0057
Clustering coefficient	0.072	0.108
Graph centralization	0.456	0.537

Table 4.7: **RDF-MTs graph metrics before and after the knowledge graph interlinking step.** Some metrics are considered for evaluating the I40KG before and after the knowledge graph interlinked step, i.e., from the number of nodes to the graph centralization. In general, results suggest an improvement in the RDF-MTs.

Figure 4.11 depicts an example of the standard RDF-MT based on the RDF molecule of the AML standard. The left side of the figure shows the basic standard RDF-MT, exemplified with the AML standard before the knowledge graph interlinking step. Figure 4.11(b) depicts eight of the new connections, i.e., properties, instances, and literal values that are incorporated to the standard RDF-MT. Figure 4.11(c) illustrates the standard RDF-MT after the knowledge graph interlinking step. In this particular case, only five properties are included for space reasons but the standard RDF-MT is extended with 35 new properties for standards. Likewise, the RDF-MT for standard organizations is enriched with 88 new properties. For example, properties such as `dbp:purpose` to represent the purpose of the organization, `dbp:leaderName`, `dbp:regionServed`, `dbp:formationYear`, are added to the RDF-MT of the standard organization. The RDF-MTs are employed to study the characteristics of the I40KG graph after the knowledge graph interlinking step (cf. Figure 4.12). The graph comprises 221 RDF-MTs and 259 intra- and inter-knowledge graph links. YAGO and DBpedia are the most utilized knowledge graphs. As expected, two RDF-MTs, i.e., `sto:Standard` and `sto:StandardOrganization` are the source of most of the generated links.

We further delve into the graph analysis of the RDF-MTs for the entire I40KG. The analysis is performed before and after the knowledge graph interlink step. The objective of this analysis is to study the I40KG with respect to the connectivity and relationships discovered in this step. Table 4.7 reports on these metrics. In total, 221 RDF-MTs with 259 links are generated based on the initial 43 (cf. Figure 4.12). Particularly, we observe that the graph density decreases, i.e., from 0.025 to 0.009 which can due to the many different types of new RDF-MTs that are added after the knowledge graph interlinked step. The clustering coefficient increases, i.e., from 0.072 to 0.108, indicating that the degree to which the RDF-MTs in I40KG graph tend to cluster together is increased in 33%. Values of transitivity are rather low and experiment a decrease, i.e., from 0.017 to 0.0057; This result can be explained by considering that while more RDF-MTs are added, they are not connected in a transitive manner. Further, values of centralization increases in 15%. As expected, the most important RDF-MTs, which are the central concepts of the I40KG are the `sto:Standard` (0.45) and `sto:StandardizationFramework` (0.41).

4.3 Discussion

Semantic interoperability conflicts are recognized as one of the core challenges in the I40 context. Standardization efforts aims at creating core terms and definitions to be used for practitioners in this context with the aim to reduce interoperability conflicts. To achieve the same goal, i.e., reduce semantic interoperability conflicts, standardization frameworks classify standards regarding their functions into different layers. Despite all these efforts, standardization frameworks, as well as standards, are not enough to resolve semantic interoperability conflicts in I40 since they are: 1) created under a specific view of a regional organization; 2) focused on particular areas and problems. A knowledge graph approach, able to encode the knowledge of standards and standardization frameworks, fosters the solution of semantic interoperability conflicts in I40 scenarios. Our approach propose a methodology to build a knowledge graph of I40 standards and standardization frameworks by examining the most relevant knowledge in this domain. Further, the proposed methodology describes the creation and refinement of a knowledge graph of standards related to I40. The objective is to semantically describe and annotate standards, as well as relations among then. In addition, the approach presented in this article allows for the semantic description of standards with respect to standardization frameworks. These semantics descriptions and annotations helps to discover new relations of standards based on the existing ones, thus, reducing interoperability conflicts. The knowledge graph internal reasoning step reveals new relations among standards. Further, the performed graph analysis is capable to discover most relevant standards, i.e., standards with the major number of connections, in the graph. The knowledge graph interlinking step is able to discover new knowledge about standards and standardization frameworks. We analyze both the number of discovered relations among standards and the accuracy of these relations. Observed results indicate that both, reasoning and linking processes enable for increasing the connectivity in the knowledge graph by up to 80%, whilst up to 96% of the relations can be validated.

We aware that our research may have two limitations. The first is that I40KG is limited to 229 samples for standards and 30 for standard organizations. A bigger number of samples could lead to a higher impact on knowledge discover. The second is that DBpedia is a general purpose knowledge graph. Therefore, domain-specific knowledge graphs could lead to a richer semantic descriptions of standards and standardization frameworks. The existence of such specific knowledge graphs are limited because of the novelty of the I40 movement and the fact these knowledge graphs are privately developed and maintained by companies of the sector.

In this chapter, we provide a semantic representation of the standard concept. Moreover, a methodology is proposed to create and refine the I40KG. The I40KG is populated with RDF molecules of standards, standardization frameworks, as well as standardization organizations. Semantic heterogeneity conflicts are resolved by means of the I40KG. Further, the reasoning and interlinking steps provide ways to refine I40KG and resolve semantic conflicts. Hence, we are convinced that I40 scenarios will benefit from the practical adoption of knowledge graphs for resolving semantic interoperability conflicts. We recommend to pay an special attention to the potential of knowledge graph-based approaches for the solution of semantic heterogeneity conflicts in I40 scenarios.

4.4 Concluding Remarks

In this chapter, a knowledge graph of I40 related standards (I40KG) is developed. We also designed the Standard Ontology (*STO*) for the semantic description of standards and their relations. Moreover, a methodology for building knowledge graphs of I40 related standards is presented. We investigated existing standardization frameworks, e.g., RAMI4.0, IIRA, IICF, and NIST. Based on these frameworks the I40KG is populated. The I40KG comprises descriptions of more than 200 standards, more than 25 standardization organizations, and 100 relations between the standards. Finally, the I40KG is linked to existing knowledge graphs such as DBpedia. An automated reasoning is implemented to reveal implicit relations between standards as well as mappings across standardization frameworks.

We analyze both the number of discovered relations among standards and the accuracy of these relations. The observed results indicate that the reasoning and linking processes enable for increasing the connectivity in the knowledge graph by up to 80%, whilst up to 96% of the relations can be validated. These outcomes suggest that integrating standards and standardization frameworks into the I40KG enable the resolution of semantic interoperability conflicts, empowering thus the *communication* in smart factories. We hope that this work contributes to a crucial step in realizing the I40 vision. The realization of this vision requires not only standards governing individual aspects, but also needs to consider semantics in the relations among standards as well as standards and standardization frameworks.

CHAPTER 5

Semantically Describing Industry 4.0 Standards Using Ontologies

The use of knowledge-based approaches in general, and semantic technologies in particular is a growing trend in the area of smart manufacturing. This development materializes in a number of approaches [72, 168, 169] that benefit from the following characteristics of semantic and knowledge representation technologies:

- formal and flexible semantic modeling with ontologies;
- intelligent, web-scale knowledge integration thanks to linked data mechanisms and ontology alignment techniques;
- browsing and exploration of distributed data sets;
- querying and reasoning based data validation and consistency checking; and
- knowledge reuse across diverse projects [86, 170].

Several areas in I40 require semantic representation for resolving heterogeneity conflicts. In this chapter, three core areas in I40 represented by standards are considered for providing a semantic representation: 1) A general architecture for I40 scenarios, i.e., RAMI4.0 and the Administration Shell concept; 2) CPS modeling by means of the AML standard; and 3) Supply Chains by means of the Supply Chain Operation Reference model. Our work aims to materialize the benefits mentioned above and support a wide-scale adoption of semantic technologies in I40 scenarios by providing a comprehensive ontology-based representation of the mentioned standards.

Problem Statement. There exists a clear gap in representing the knowledge of standards in I40 scenarios by means of ontologies and knowledge graphs.

This chapter addresses the second level in the general proposed solutions, i.e., semantically representing Industry 4.0 standards with ontologies (cf. Figure 5.1). Particularly, the research question addressed in this chapter is as follows:

RQ2: Can knowledge graphs represent semantics encoded in Industry 4.0 entities?

Proposed solution. To meet the need of representing ontologies in I40 scenarios, we derive a practical methodology. The methodology is used to build ontologies of important standards

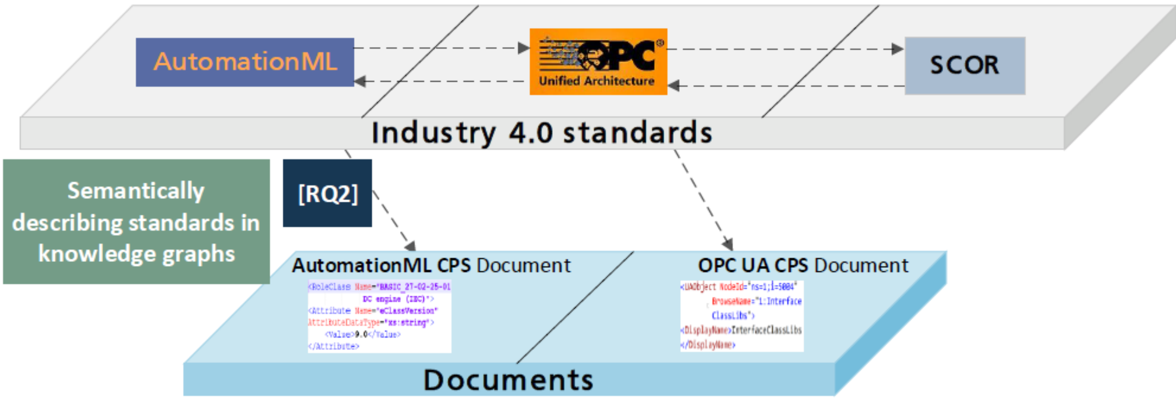


Figure 5.1: **The levels of the research problem addressed in this chapter.** We address challenges concerning the semantic representation of Industry 4.0 related standards.

in the domain. Finally, we showcased the utilization of the ontologies to resolve semantic heterogeneity conflicts in I40 scenarios. Particularly, we present the following contributions which respond to **RQ2**:

- A practical methodology for building ontologies for I40 related standards;
- The *RAMI4.0* ontology[1]: covering the RAMI4.0 specification and allowing the semantic representation of the RAMI4.0 architecture as well as the Administration Shell concept;
- The *AML* ontology: representing the AutomationML standard[2];
- The SCORVoc vocabulary[3] for semantically describing the Supply Chain Operation Reference; and
- Use cases showing the applicability and benefits of using semantic representations of standards in I4.0 scenarios.

This chapter is based on the following publications [171–175]. The remainder of this chapter is structured as follows. In Section 5.1, the methodology used for developing the ontologies is presented. Then, Section 5.2 presents the RAMI4.0 ontology, Section 5.3 outlines the AML ontology, and Section 5.4 the SCORVoc vocabulary. In addition, uses cases demonstrating the applicability of each of the presented ontologies are outlined. Concluding remarks for this chapter are drawn in Section 5.5.

5.1 Methodology

In order to design the ontologies, the "process-based design" methodology of Uschold *et al.* [176] is adapted. Further, the *VoCol* methodology and support environment [177] are also considered to develop the ontologies. VoCol is a platform to support the collaborative development process of ontologies based on version control systems, Git and GitHub in this case. VoCol, after

[1] https://w3id.org/i40/rami#
[2] https://w3id.org/i40/aml#
[3] http://purl.org/eis/vocab/scor#

every push to the GitHub repository, automatically provides features such as documentation generation, the evolution of changes, visualization and ontology validation. As the main purpose for ontology creation the following points are formulated:

1. **Define the purpose and scope.** Defining the purpose of the ontology, i.e., to semantically model the standard or a representative part of it.
2. **Capture the domain knowledge.** Investigate and analyze existing ontologies (if any) for the standard; if no ontologies exist for the standard, the domain knowledge is typically extracted from documents, e.g., technical standard specifications, white papers, scientific articles.
3. **Develop the ontology.** Develop the ontology by focusing on the core classes. A comprehensive diagram of the ontology, as well as a description of the core classes and relations, is required.
4. **Consider best practices for developing ontologies of I40 standards.** Best practices for developing ontologies with respect to reuse, documentation, naming conventions, multilinguality, and availability need to be observed (cf. Section 5.1.1).
5. **Evaluate the applicability of the ontology with use cases.** Concrete instantiation and application of the ontology are evaluated through use cases. Use cases are required to reflect general problems using standards in I40 scenarios as well as the benefits of employing ontologies in this area.

5.1.1 Best Practices for Developing Ontologies of I40 Standards

In an attempt to ensure a certain level of adequacy of the ontologies, a set of best practices for ontology development are proposed [174]. The major focus is on performing all necessary steps to ensure high-quality documentation and availability of the ontology for other interested parties, thus, facilitating ontology reuse.

In the following, a comprehensive list of best practices for ontology development is presented. This list is derived from our own experience in creating industrial ontologies as well as from the results of an analysis of widely-used ontologies [174]. These practices serve as guidelines that help to focus on the most important aspects of the ontology development process. Therefore, by using these practices, it is expected to increase the efficiency of the collaboration and to improve the overall quality of the ontologies.

Reuse

Currently, in ontology development, the reuse of existing terms is an aspect of vital importance [178, 179]. The main idea is not to create new terms but to utilize those that are present in the existing ontologies and to avoid redundant work. Apart from saving time and investment costs, ontology reuse is expected to ensure a certain level of quality. The reason for this is that the longer an ontology exists and is reused, the more review processes it has gone through. Additionally, according to [180] reuse is considered to be a best-practice in ontology development. Therefore, in the following, important practices regarding reuse are discussed.

P-R1 Reuse of Ontology Design Patterns Ontology design patterns are reusable modeling building blocks providing solutions to recurrent domain modeling problems [181]. ODPs are an important means to improve the quality of an ontology design as they represent best practices in ontology modeling frequently used by ontology developers. Sabou *et al.* [86] distinguished three major groups of patterns that are important in smart manufacturing contexts: a) part-whole relations; b) connections between components; and c) component roles. Based on these criteria, some of the ODPs that are more commonly utilized to model standards are outlined.

- **Part-whole** relations are important for modeling containment hierarchies. The *PartOf ODP*[4] pattern allows to represent entities and their parts with transitivity.
- **Constituency** refers to relations without a clear part-of relationship. A typical example is representing a material from which an object is made, e.g., several types of wood constitute a table. There is a special ODP defined for modeling constituency - *Constituency ODP*[5].
- **Componency ODP** models non-transitively that objects either are proper parts of other objects or have proper parts[6] (non-transitive version of part-whole).
- **TimeIndexedPartOf ODP**[7] represents part-whole relations which holds only for a specific time interval.

P-R2 Reuse of well-known ontologies We considered well-known ontologies as ontologies which are: (1) published by renowned standardization organizations; (2) widely used in a large number of other ontologies; (3) defined in a more domain independent way addressing more general concerns; and (4) comprise relevant concepts for I40 scenarios. Reusing well-known ontologies increase the probability that data can be consumed by applications [182]. Hence, we propose these most widely used ontologies as the first option for reuse. Table 5.1 depicts some of the general ontologies that are of importance for I40 scenarios. Thus, these ontologies are proposed to be surveyed when building a new ontology in this settings.
P-R3 Representing units of measurement Units of measurement are of paramount importance in I40 scenarios for the correct function and coordination of processes. In this regard, we researched and tested the existing implementation of ontologies covering this knowledge. The Ontology of Unit of Measurements (OM) [183] provides a complete and well documented implementation for describing units, quantities, measurements, and dimensions. Based on this, we propose to use this ontology to represent this type of knowledge.
P-R4 Avoid semantic clashes If the term has a *strong* semantic meaning for the domain, different from the existing ones, then a new element needs to be created.
P-R5 Individual resource reuse Especially, elements from well-known ontologies are proposed to be reused as individual ontology elements.
P-R6 Vocabulary module reuse (Opposite to **P-R4**) Often ontologies require certain basic structures such as addresses, persons, organizations, which are already defined in existing ontologies. Usually, such structures comprise the definition of one or several classes and a number of properties. If the conceptualizations match, the complete reuse of a whole module needs to be considered.
P-R7 Establishing alignments with existing ontologies Instead of the strong semantic commitment of reusing identifiers, alignments using properties such as `owl:equivalentClass`, `owl:equivalentProperty`, `rdfs:subClassOf`, and `rdfs:subPropertyOf` can be established.

[4] http://ontologydesignpatterns.org/wiki/Submissions:PartOf
[5] http://ontologydesignpatterns.org/wiki/Submissions:Constituency
[6] http://ontologydesignpatterns.org/wiki/Submissions:Componency
[7] http://ontologydesignpatterns.org/wiki/Submissions:TimeIndexedPartOf

Table 5.1: **Well-known ontologies for Industry 4.0 scenarios.** Relevant ontologies of general purpose that can be applied in Industry 4.0 scenarios.

Name	Prefix	Domain
Friend Of A Friend `http://xmlns.com/foaf/0.1/`	*foaf*	Terms related to Persons (i.e., Agent, Document, Organization, etc).
Dublin Core ontology Terms `http://purl.org/dc/terms/`	*dcterms*	General metadata terms (i.e., Title, Creator, Date, Subject, etc).
Simple Knowledge Organization System Namespace `http://www.w3.org/2004/02/skos/core#`	skos	Data model for sharing and linking knowledge organization systems.
Vocabulary of Interlinked Datasets `http://rdfs.org/ns/void#`	*void*	Metadata about RDF datasets (i.e., Dataset, Linkset, etc).
Provenance Ontology `http://www.w3.org/ns/prov#`	*prov*	Provenance data model (i.e., Entity, Activity, Agent).
Ontology of Units of Measurements `http://www.ontology-of-units-of-measure.org/page/om-2`	*om*	Represents units of measurements (i.e., Unit, Quantity, Measurement, and Dimension).
Semantic Sensor Network Ontology `http://www.w3.org/ns/ssn/`	*ssn*	Represents Sensor, actuators, and observations (i.e., Observation, Stimulus, Platform, etc).
WGS84 Geo Positioning `http://www.w3.org/2003/01/geo/wgs84_pos#`	geo	Represents longitude and altitude information in the WGS84 geodetic reference datum.
Socially Interconnected Online Communities ontology `http://rdfs.org/sioc/ns#`	*sioc*	Aspects of online community sites (i.e., Users, Posts, Forums, etc).
Time Ontology `http://www.w3.org/2006/time#`	*time*	Time information (i.e., Duration, Day, Time Intervals, etc).
Data Cube Vocabulary `http://purl.org/linked-data/cube#`	*qb*	Statistic data (i.e., Dimensions, Attributes, Measures, etc).
Description of a Project `http://usefulinc.com/ns/doap#`	*doap*	Terms for Open Source Projects (i.e., Version, Repository, etc).
Bibliographic Ontology `http://purl.org/ontology/bibo/`	bibo	Citations and bibliographic references (i.e., Quotes, Book, Article, etc).
Data Catalog Vocabulary `http://www.w3.org/ns/dcat#`	dcat	Facilitate interoperability between data catalogs published on the Web.
Schema.org `http://schema.org`	*schema*	Broad schema of concepts (i.e., Event, Organization, Person, etc).
GoodRelations `http://purl.org/goodrelations/v1`	gr	E-Commerce related terms (i.e., Product, Service, Location, etc).
Creative Commons schema `http://creativecommons.org/ns`	*cc*	Describes copyright licenses (i.e., License Properties, Work Properties, etc).
GeoNames `http://www.geonames.org/ontology`	gn	Geospatial semantic information (i.e., Population, PostalCode, etc).
DUL ontology `http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#`	dul	Upper ontology (i.e., Entity, Object, Agent, etc).
Event Ontology `http://purl.org/NET/c4dm/event.owl`	event	Describes reified events (i.e. Event, Location, Time, etc).

Documentation

Providing a user-friendly view of vocabularies for non-experts is crucial for integrating Semantic Web with everyday Web [184]. It facilitates the contribution of domain experts during the development process. In addition, it helps other interested parts for easy use of the ontology in later phases as well. There exist different tools for documentation generation. Basically, these tools require to include the following information for each resource to provide a basic documentation.

P-Do1 Use of rdfs:label and rdfs:comment To this end, we propose adding basic documentation for every element, i.e., `rdfs:label` or `skos:prefLabel` and describing the meaning of the element in natural language by using `rdfs:comment` or `skos:definition`.

P-Do2 Generate human-readable documentation Easy-to-use documentation is critical for the wide adoption of the vocabulary.

P-Do3 Reference the sources for the ontology elements When creating an ontology for standards, it is typically to base the work on existing white papers, standard specifications, and technical reports. Using `rdfs:isDefinedBy` to describe the resource (s) helps to maintain the ontology. Furthermore, this enables the understanding of the concepts that are defined in the current version of the ontology.

Naming Conventions

Following naming conventions has a high impact on vocabulary development [185]. Naming conventions help to avoid lexical inaccuracies and increase the robustness and exportability, specifically in cases when ontologies are aligned with external ontologies [182]. The utilization of meaningful names increases the robustness of context-based text mining for automatic term recognition and ease the manual and automated integration of terminological artifacts, i.e., comparison, checking, alignment and mapping [185, 186]. Considering the literature on this topic [182, 187] the following practices are proposed. For ontology construction, the use of the *CamelCase* notation is considered as a best practice [188]. Therefore, we propose the use of this specific notation.

P-N1 Concepts as single nouns Name all concepts as single nouns using CamelCase notation, e.g., *PlanReturn*.

P-N2 Properties as verb senses Name all properties as verb senses also following CamelCase approach. To clearly distinguish from class names, the name of a property is required to be a plain noun phrase, e.g., `hasProperty` or `isPropertyOf`.

P-N3 Short names Provide short and concise names for elements. When natural names contain more than three nouns, use the `rdfs:label` property with the long name and a short name for the element. For instance, for `ManageSupplyChainBusinessRules` use `BusinessRules` and set the full name in the label. In order to explain the context, e.g., Supply Chain, complement this label with the `skos:altLabel`.

P-N4 Logical and short prefixes for namespaces Assign logical and short prefixes to namespaces, preferable, with no more than five letters, i.e., `rami:XXX, aml:XXX, scor:XXX`. To describe the ontologies, we utilize the notation prefix:element; prefix refers to the identification of the ontology and element can point to a class, a property or an instance of the ontology.

P-N5 Regular space as word delimiters for labeling elements Add descriptions for terms that follows the normal writing of sentences, i.e., with regular spaces between words. For example, `rdfs:label "A Process that contains.."`.

P-N6 Avoid the use of conjunctions and words with ambiguous meanings Avoid names with “And”, “Or”, “Other”, “Part”, “Type”, “Category”, “Entity” and those related to datatypes like “Date” or “String”.

P-N7 Use positive names Avoid the use of negations. For instance, instead of `NoParkingAllowed` use `ParkingForbidden`.

P-N8 Terminology Respect the terminology used by standards, standardization frameworks, registered products, and company names. In these cases, the use of CamelNotation is not recommended. Instead, the name of the standard, standardization frameworks, registered products, or company requires to be used as is, e.g., OPC UA, IEC 62714, Daimler AG. The main intention is to facilitate the understanding of the ontology constructs and their semantics for users that are already familiar with the standards but might not possess deep knowledge of semantic technologies.

Multilinguality

Providing multilingual ontologies is desirable but not a straightforward issue [189]. In the following, we propose some practices to be observed.
P-M1 Use English as the main language Use English to annotate all the elements in the ontology; explicitly include the *@en* notation.
P-M2 Multilinguality Every element in the ontology is required to contain at least one annotation in the English language. Additionally, to add another language, complement the object with the multilingual annotation.

Availability

The availability comprises practices for the ontology to be used and improved. To this end, the following practices are proposed.
A-P1 Publishing at a persistent URI. The ontology needs to be published under a persistent URI. The W3ID service[8] provides means to accomplish this requirement.
A-P2 License specification. The ontology is required to contain the definition of a license specifying to which extend it can be reproduced.
A-P3 Available ontology sources. As a best practice, publishing the sources of the ontology in a public service, e.g., LOV or Github to foster development and collaboration of the ontology.

5.2 The RAMI4.0 Ontology

In this section, we present the RAMI4.0 ontology. This ontology is built based on the specification of the RAMI4.0 model. The RAMI4.0 model depicts a general architecture describing the dimensions and layers of I40 scenarios. Further, it comprises the Administration Shell concept which aims to represent assets along the complete product life-cycle. An ontology-based representation of the RAMI4.0 model and the Administration Shell permit the following improvements:

- flexible schema representations characterizing standards which participate in all the dimension and layers of RAMI4.0;
- using the semantic web and linked data technologies to validate and generate new knowledge for I40 scenarios; and
- semantic descriptions of the assets throughout their life-cycle allowing to resolve semantic interoperability conflicts in I40 scenarios.

The RAMI4.0 specification is a rather new standard. Thus, there exist few works aiming to represent the semantics in this domain by means of ontologies [93, 95, 96].
However, these works present the following drawbacks:

1. only cover the RAMI4.0 and Administration Shell concepts;
2. not developed according to best practices of ontology design;
3. not available online for consulting or extending; and

[8] https://w3id.org/

4. tailored for specific use cases.

To overcome these drawbacks, the RAMI4.0 ontology outlined in this section aims to represent the RAMI4.0 specification as well as the Administration Shell concept [6, 190]. In the following, specific requirements of I40 and how they are addressed are presented. Additionally, the methodology proposed in this chapter for creating ontologies in I40 scenarios is used for developing the RAMI4.0 ontology.

5.2.1 Challenges for Realizing Industry 4.0

As explained in Section 2.1.2, the RAMI4.0 model describe a general architecture for solutions in I40 scenarios. Furthermore, the Administration Shell depicts a smart interface to an asset. Additionally, some of the existing challenges for the development of I40 are presented in detail.

Global unique identification (Ch1) Enabling intercommunication among I40 components and the environment over the Internet is a big challenge. In addition, a linking mechanism among I40 components and the data they generate is required [191]. Therefore, addressing this challenge is crucial for the realization of the vision of I40.

Data availability (Ch2) Another challenge is the availability of data beyond the boundaries of the manufacturers and across different hierarchy levels. This challenge becomes even harder when various policy rules from manufacturers are applied. I40 components communicate with each other and interact with the environment through exchanging data generated from different sensors and react to the events by triggering actions with the aim of controlling the physical world [192]. Therefore, sharing generated data between participants is a key factor for I40 [193].

Standardization compliance (Ch3) Standardization processes is an important step toward the realization of I40. Several standards to deal with different layers in factories exist nowadays. For instance, *AutomationML* [194], *Profibus* [195] and *OPC-UA* [196, 197] are just some of the examples of the mentioned standards. The core idea of all this effort is to provide a detailed description of the components in manufacturing processes. Manufacturing processes constantly generate different components and standards needed to reflect this dynamically.

Multilinguality (Ch4) In order to achieve a wide range of applicability to different cultures and communities [198], localization and internationalization are required. This permit decreasing the learning curve and allow easier and faster adoption of I40 in real production environments.

Lack of a Standard Architecture Model for Industry 4.0 solutions (Ch5) One the aims of the RAMI4.0 model is to provide an architecture to frame I40 oriented solutions. However, other standardization frameworks have been developed in other regions, e.g., USA [22], China [49, 80], just to mention a few. Despite this, RAMI4.0 represents a step towards the realization a standard architecture model for I40 solutions. Current efforts of different communities aim to leverage interoperability between existing frameworks such as RAMI4.0 and IIRA – IICF [8].

The listed challenges are some of the many existing ones for achieving the I40 vision. Providing a semantic representation of the RAMI4.0 model as well as its Administration Shell concept aims to solve these challenges. Further, the semantic representation enables the solution of interoperability conflicts in this domain.

5.2.2 Methodology

In this section, we apply the methodology proposed in this chapter, to build the RAMI4.0 ontology. General best practices are observed and specific design decisions are taken to cover particularities of the RAMI4.0 domain and the Administration Shell. Table 3.1 reports on existing state of the art approaches for the semantic representation of assets in I40 scenarios. The works are examined with respect to the basic concept that they represent, how the identification is performed, the data model, as well as the formalization. This analysis is utilized for the development of the RAMI4.0 ontology.

5.2.3 Purpose and Scope

It is widely accepted that semantics technologies play a crucial role in the management of things, devices, and services [199, 200]. Moreover, RAMI4.0 [20] recognizes as a requirement that I40 components and their contents follow a common semantic model. Therefore, we propose a semantic approach to address the challenges presented in section 5.2.1. The purpose of the RAMI4.0 ontology is to provide a semantic representation of the RAMI4.0 model as well as for the Administration Shell concept (cf. Figure 2.2).

5.2.4 Capture

The capture of the domain mainly follows the description of the RAMI4.0 core description documents [6, 20]. The RAMI4.0 ontology focuses on describing its three dimensions, i.e., Hierarchy Level, IT, and Product Life-Cycle. Additionally, we consider the layers of the dimensions where the integration of data in I40 scenarios is made (cf. Section 2.1.2).

Further, the Administration Shell concept, that provides a digital representation of all information being available about and from an asset is of core relevance for the RAMI4.0 ontology. Assets are viewed as industrial devices, ranging from simple components to complex CPS. Assets exhibit different capabilities, e.g., communication or execution of production steps. These capabilities may change during the individual life cycles. Further, the Administration Shell provides methods for the interaction with assets, as well as capabilities of an asset that are required for its employment. Hence, an Administration Shell essentially depicts a *smart interface* to an asset. This smart interface may vary in its asset-specific extent but always provides a standard access point for knowledge discovery and utilization of assets.

Global unique identification Identification of each I40 component by using global unique identifier ensures entity disambiguation and retrievable [201]. According to Linked Data principles [40], HTTP URIs are recommended to be used for naming things. Following this principle, we propose that each I40 component need to be identified by an HTTP URI. By doing so, a decentralized, holistic and extensible global unique identification scheme for I40 components is established. As a consequence, I40 components become dereferenceable able to self-locate and communicate with each other. Listing 5.1 presents our proposal for identifying the I40 components. In addition, it shows that identification capabilities can be extended by various existing vocabularies that provide adequate means. This example uses the term `identifier` from the *Dublin Core Vocabulary*[9] to achieve a reference to the resource which is not ambiguous within a given context.

[9] http://dublincore.org/documents/dcmi-terms/#H1

```
@prefix rami: <http://w3id.org/i40/rami>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix dcterms: <http://purl.org/dc/terms/>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

#Class definition
rami:Actuator       a owl:Class;
    rdfs:subClassOf  rami:Asset;
    rdfs:label       "Actuator"@en;
    rdfs:comment     "Actuator is a component of a machine that..."@en;
    rdfs:isDefinedBy  <https://en.wikipedia.org/wiki/Actuator>.

#Property reusing
rami:hasAssetId a owl:DatatypeProperty ,
                  owl:FunctionalProperty;
    rdfs:subPropertyOf  dcterms:identifier ;
    rdfs:label     "has Asset Id"@en ;
    rdfs:comment   "Unique identification for the Asset."@en ;
    rdfs:domain    rami:Asset ;
    rdfs:range     xsd:string .

#Instance definition
rami:ActuatorAAA001 a rami:Actuator;
    rdfs:label       "Actuator ID AAA001"@en;
    rami:hasAssetId  "AAA001"^^xsd:string .
```

Listing 5.1: **Global ID with RDF.** Global identification of properties is achieved by reusing a well-known ontology, i.e., DCTERMS and adding more sophisticated restrictions, e.g., `owl:FunctionalProperty`. In this way, the semantic of an identifier is used and improved in the local ontology.

In particular cases, the property `dcterms:identifier` is utilized by defining a subproperty. Here, more sophisticated restrictions need to be added, e.g., `owl:FunctionalProperty`.

Data availability The benefits of employing RDF as the standard for representation of the data are twofold. Firstly, various data serialization formats are easy to be generated and transmitted over the network. Secondly, using SPARQL, as a W3C Recommendation for an RDF query language, it is possible to make data available through a standard interface. RDF representation of data can be created on the fly, even if data are stored in relational databases or other data formats [202]. In such a case, a semantic-based approach enables data sharing between legacy systems and other participants in a networked manufacturing as well.

Standardization compliance Following the idea of employing RDF as a lingua franca for describing data in I40 scenarios, we propose to translate existing standards into ontologies. Standards are used to categorize submodels. Submodels are part of the Administration Shell and are utilized to represent the different areas of an asset, e.g., identification, communication, energy efficiency (cf. Figure 2.2(b)). The interoperability between standards can thus be managed through the integration of the respective ontologies. In addition, these ontologies are also connected with the Administration Shell.

Multilinguality Since various communities across the world interact with I40 components, it is very important that they can interact with terms in their own language. Semantic web technologies enable implementation of multilinguality in a very straightforward manner. This remains valid even for the newly introduced languages or concepts. The following example illustrates this practice with multilingual annotations, i.e., English and German for the class `rami:AdminShell`.

```
@prefix rami: <http://w3id.org/i40/rami#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

rami:AdminShell a owl:Class ;
    rdfs:label   "Admin Shell"@en,
                 "Verwaltungsschale"@de;
    rdfs:comment "Describes the Administration Shell concept..."@en,
                 "Beschreibt das Konzept der Verwaltungsschale..."@de.
```

Listing 5.2: **Multilinguality example for the AdminShell class.** Multilingual annotations in English (@en) and German (@de) are added to the class AdminShell by means of the `rdfs:label` and `rdfs:comment` properties.

5.2.5 Design

An ontology for RAMI4.0 comprising the Administration Shell as a core is provided in this work. The Administration Shell concept provides a semantic description of the elements that describe I40 components. Since the Administration Shell is a key concept of the RAMI4.0 model, we decided to use the namespace *rami* for the ontology—also, since the ontology implements further concepts of the RAMI4.0 model.

Ontology Overview

In this section, we describe the main classes of the RAMI4.0 ontology (cf. Figure 5.2).

rami:AdminShell represents the Administration Shell concept and its properties.

rami:Asset Assets in RAMI4.0 are described by the `rami:Asset` class. In addition, properties like `rami:hasAssetID`, `foaf:image`, and `dul:isPartOf`, are created to represent characteristics of the asset.

rami:I40Component this class depicts the concept of the I40 component. An I40 component characterizes the physical structure of an asset. A given Asset described by an Administration Shell can be considered as an I40 component. I40 components will communicate among each other in I40 environments.

rami:Submodel models the submodel concept in the Administration Shell. The basic data associated with the asset are represented by `rami:Submodel` class. The classes `rami:EnergyEfficiency`, `rami:Structure`, `rami:MESConnection` allows to model different types of data, attached to the Administration Shell concept, as subclasses of the `rami:Submodel` class.

rami:Dimension this class describes the three different dimensions of RAMI4.0, i.e., the hierarchy level, the IT, and the life-cycle. The three dimensions are represented as subclasses of the `rami:Dimension`. Moreover, an instance representing every layer in the dimensions is added to the respective classes. For example, `rami:Enterprise` is added as an instance of the class `rami:HierarchyLevelDimension` to represent the enterprise level of this dimension. Likewise, the classes `rami:View`, `rami:ExpressionSemantic` are populated with instances representing the possible values that they can have. To express this knowledge, the `owl:oneOf` construct is used. For example, the semantic expressions in RAMI4.0 can only be `rami:Confirmation`, `rami:Measurement`, and `rami:Requirement` [203]. These concepts are created as instances of the `rami:ExpressionSemantic`. Then, they are modeled as the only instances that this class can have by means of `owl:oneOf`. This way of modeling is followed in all the classes where the instances can belong to a restricted set of individuals.

Description of Properties

In this section, the core properties of the RAMI4.0 ontology are outlined.

rami:comprises this property connects two of the main classes of the ontology, i.e., the `rami:AdminShell` and the `rami:SubModel`.

rami:isConnectedTo links the Administration Shell concept with the different dimensions described by the RAMI4.0 model.

rami:containsAdminShell describes the connection between the Administration Shell concept and the I40 component.

rami:describesAsset associates the Asset with the Administration Shell.

rami:inAccordanceWith this property outlines the link between a certain standard and a given submodel. The standard is described by the *sto:Standard* class from the *STO* ontology (cf. Section 4.1.1). This approach allows connecting the asset with a given standard that describes it via the Administration Shell [204].

rami:hasDatatype Data types are important for the description of the submodels. This property points to the `rdfs:Datatype` class to model the connection between a given data type that is used to describe a certain submodel.

rami:hasManifest this property relates the concepts of the Administration Shell and its correspondent Manifest.

Reusing Well-known Ontologies

Some well-known ontologies are reused to represent the domain of RAMI4.0 and the Administration Shell concept. The OM ontology is utilized to model the units of measurements. For instance, the class `om:Unit` as well as the property `om:hasUnit` are used with the `rami:Submodel` class to represent the units of measurements which are required properties of the submodels. The SSN ontology is exploited to represent data about sensors. To this end, the classes `ssn:Sensor` and `ssn:Property` are linked to the submodel `rami:SensorMeasurement` by using the `ssn:isProducedBy` property. The PROV ontology is reused by means of the class `prov:Activity` to describe the activity that generated a specific Administration Shell (`rami:AdminShell`). In this case, the object property, `prov:wasGeneratedBy` links the two classes. The FOAF ontology and the SKOS vocabulary are utilized to represent connections of the classes `rami:Asset` and `rami:Submodel` respectively. The `foaf:image` property models the image of the asset while the `skos:definition` depicts the definition of the Submodel. The identifiers in the RAMI4.0 ontology are added as subproperties of the `dcterms:identifier` property, e.g., `rami:hasAssetId` and `rami:hasAdminShellId` are subproperties of `dcterms:identifier`.

Ontology Design Patterns

We examine the characteristics of the RAMI4.0 model and the Administration Shell to determine which ODP can be reused. Since an asset can be part of other assets and is of interest to obtain the transitivity in this relation, the PartOf ODP is utilized here. Additionally, to encode the relation between the `rami:AdminShell` and the `rami:ComponentManager` we employed the Componency ODP. This ODP is expressed through the property `dul:hasComponent` to express that the `rami:ComponentManager` is a proper part of a `rami:AdminShell`.

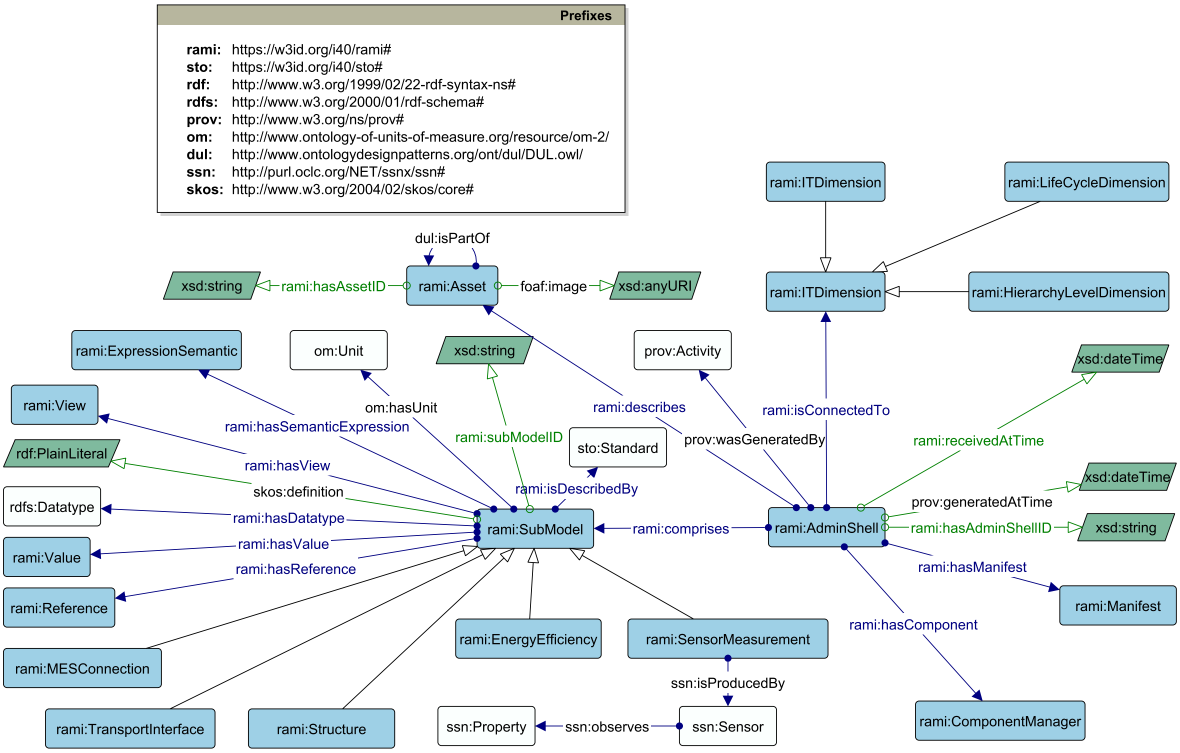


Figure 5.2: **Core classes and properties of the *RAMI4.0* ontology.** Core concepts related to the RAMI4.0 and the Administration Shell are represented in the diagram. Blue squares depict classes, blue arrows depict object properties and green arrows datatype properties of the RAMI4.0 ontology. Reused classes from well-known ontologies are depicted in white, e.g., SSN, PROV, and OM. Reused properties are shown in black, e.g., DUL, FOAF, SKOS. Green squares show used datatypes.

Units of Measurements

Units are required for specification of products as well as for representing data produced by measuring devices, e.g., sensors. Typically, units of measurements are represented as simple strings, e.g., °C, mm, kg, etc. In such cases, the meaning of units of measurements is not machine-readable and sometimes unknown or ambiguous. For example, both “18 in” and “45,72 cm” are referring to the same length. For properly representing units, an alignment of the *RAMI4.0* ontology with the OM ontology is proposed. By using the *in*[10] and *cm*[11] concepts from the OM ontology, the semantics of units can be *understood* by a machine because their formal definitions can be looked up in the ontology via the IRIs of the concepts as well as processed and interpreted by software. For example, “centimetre” is defined as a unit in the dimension of length, amounting to 1/100 of the SI unit “metre”. Listing 5.3 illustrates how data values can be represented using the OM ontology and how semantic interoperability conflicts can be addressed.

[10] http://www.ontology-of-units-of-measure.org/page/om-2/inch-International
[11] http://www.ontology-of-units-of-measure.org/resource/om-2/centimetre

```
@prefix owl:  <http://www.w3.org/2002/07/owl#> .
@prefix om:   <http://www.ontology-of-units-of-measure.org/resource/om-2/> .
@prefix rami: <http://w3id.org/i40/rami/> .

rami:asset1              a rami:Asset;
    rami:hasLength       rami:lengthOfAsset1 .
    rami:lengthOfAsset1  om:hasNumericalValue  "45.72"^^xsd:float ;
                         om:hasUnit om:centimetre .

rami:asset2              a rami:Asset;
    rami:hasLength       rami:lengthOfAsset2 .
    rami:lengthOfAsset2  om:hasNumericalValue  "18"^^xsd:float ;
                         om:hasUnit om:inch-International .

rami:lengthOfAsset1  owl:sameAs  rami:lengthOfAsset2.
```

Listing 5.3: **Representing units of measurements with the OM ontology.** An example using different units of measurements, i.e., centimeter and inches, to represent the same length of an asset. This representation enables the resolution of the semantic interoperability conflicts of Domain (**SIC3**) and Representation (**SIC4**).

5.2.6 Summary of Ontology Characteristics

While designing RAMI4.0, best practices for ontology development are presented in Section 5.1.1. The documentation of the ontology is publicly available via a *VoCol* instance.[12] The RAMI4.0 ontology is published at the w3id[13] and is published under the Creative Commons license. The source code for the RAMI4.0 ontology as well as its evolution track is publicly available on GitHub.[14] The reuse of well-known ontologies and the use of ODPs are included in the development process of the RAMI4.0 ontology. To ensure that the RAMI4.0 is self-explanatory we use the following properties for all terms: a) `rdfs:label` to include the most commonly used name for ontology entities b) `rdfs:comment` to explain the meaning of main entities; b) `skos:altLabel` to include the alternative names for ontology entities. Table 5.2 summarizes the main characteristics of the RAMI4.0 ontology.

5.2.7 Use Cases

The vision of I40 is centered around the concept of decentralized production and smart objects that participate in the production in terms of autonomy and decision-making. To accomplish this goal, asset metadata, data, and relations with other assets need to be semantically described with the *RAMI4.0* ontology. For this purpose, the information provided by one asset can be understood and exploited by other smart objects in the production chain. Next, two uses cases showing the applicability of the RAMI4.0 ontology are described. The first one shows how the ontology can be used to query a legacy sensor data. The second use case demonstrates how the ontology can be utilized to model and resolve semantic interoperability conflicts in this domain.

Legacy Sensor Data

To illustrate the applicability of the RDF-based approach, a use case is detailed where the *RAMI4.0* ontology is used to describe data of a legacy system and some of its basic relations.

[12] http://vocol.iais.fraunhofer.de/rami/
[13] https://w3id.org/i40/aml#
[14] https://github.com/i40-Tools/RAMIOntology

Table 5.2: **Summary of the RAMI4.0 Ontology Characteristics.** The table shows a summary of the RAMI4.0 ontology in aspects such as general details, reused ontologies, documentation, naming conventions, multilinguality, availability, and used methodology.

General	**Name:**	RAMI4.0 Ontology (rami)
	Size	30 classes, 22 object properties, 33 data properties, 21 individuals
	DL Expressivity	SHOF(D)
Reuse	**P-R1**, **Reused Ontologies**	PROV, OM, SSN, DUL, FOAF
	P-R2, **Reused ODPs**	transitive PartOf ODP
Documentation	**P-Do1**, **P-Do3**	By means of `rdfs:label`, `rdfs:comment`, `skos:prefLabel` and `rdfs:isDefinedBy`
Naming Conventions	**P-N1** – **P-N8**	CamelCase notation
Multilinguality	**P-M1**, **PM2**	English for all terms and german labels for core elements
Availability	**PersistentURI**	https://w3id.org/i40/rami#
	GitHub	https://github.com/i40-Tools/RAMIOntology
	LOV	http://lov.okfn.org/dataset/lov/vocabs/rami
	Licence	Creative Commons
	VoCol Instance	http://vocol.iais.fraunhofer.de/rami/
Methodology	**Ont. Eng. Methodology**	King and Uschold [176]

Table 5.3: **Mapping dataset columns of the AirProbe with RAMI4.0 properties.** The columns in the AirProbe dataset are mapped to the concepts represented by properties of the RAMI4.0 ontology.

Column from AirProbe DB	*Vocabulary concept*
meta_device_id	rami:hasAssetId
meta_timestamp_recorded	prov:generatedAtTime
meta_timestamp_received	rami:receivedAtTime
geo_lat	geo:lat
geo_lon	geo:long
...	...
data_temp_1	om:Temperature
data_hum_1	om:RelativeHumidity

To accomplish this objective, the *AirProbe* dataset is utilized [205]. This dataset is provided as a SQL dump and comprises data about sensors, their geospatial locations, measurements of black carbon concentrations, temperature, and humidity. Such types of data are typically present in industry contexts, for instance, if sensors are installed in a factory, machine, or carrier. Sensor data is modeled in the *RAMI4.0* ontology by the `rami:SensorMeasurement` class.

To support the mapping of the datasets to the *RAMI4.0* ontology we use *R2RML*[15], the W3C standard for mapping relational databases to RDF datasets. The mappings between the properties of the ontology and the dataset are generated with the *D2RQ* tool [146].[16] Table 5.3 shows an excerpt of the mappings between ontology and the dataset. D2RQ is used on top of a MySQL server to make the dataset accessible as RDF. D2RQ acts as a middle layer between the SQL-based data and the SPARQL queries. As a result, it is possible to perform queries and to receive real-time information about particular events. In this way, the data in a legacy system can be used and exploited by other, RDF-aware software agents without the need to transform all of them into RDF following an ETL (extract-transform-load) approach, which is expensive if the data source is updated frequently. Listing 5.4 shows the query used to obtain the measured temperature for a specific time interval. The result of the query is depicted in Figure 5.3.

[15] http://www.w3.org/TR/r2rml/
[16] http://d2rq.org

```
PREFIX rami: <https://w3id.org/i40/rami#>
PREFIX xsd:  <http://www.w3.org/2001/XMLSchema#>
PREFIX om:   <http://www.ontology-of-units-of-measure.org/resource/om-2/>


SELECT  ?time  ?value  WHERE {
    ?measurement  rami:receivedAtTime  ?time ;
                  om:Temperature       ?value .
    FILTER  (xsd:dateTime(?time) >= "2015-01-29T10:00:00Z"^^xsd:dateTime)
    FILTER  (xsd:dateTime(?time) <= "2015-01-29T11:00:00Z"^^xsd:dateTime)
}
ORDER  BY  DESC(?value)  LIMIT  20
```

Listing 5.4: **Querying temperature in a time interval of one hour.** Retrieving values of the temperature of a sensor of the AirProbe dataset in a time interval of one hour by using the RAMI4.0 ontology.

The query defined in Listing 5.5 returns all geospatial information about those sensors that transmitted data in a particular interval of time. Figure 5.4 shows these geographical coordinates on an interactive map, where the user is able to navigate and obtain more information about the sensors. Listing 5.4 and Listing 5.5 present SPARQL queries that illustrate the data retrieval possible with this approach. The use case points out how the *RAMI4.0* ontology enables a flexible semantic representation of data, which helps to overcome the challenges related to the integration of heterogeneous data sources in I40.

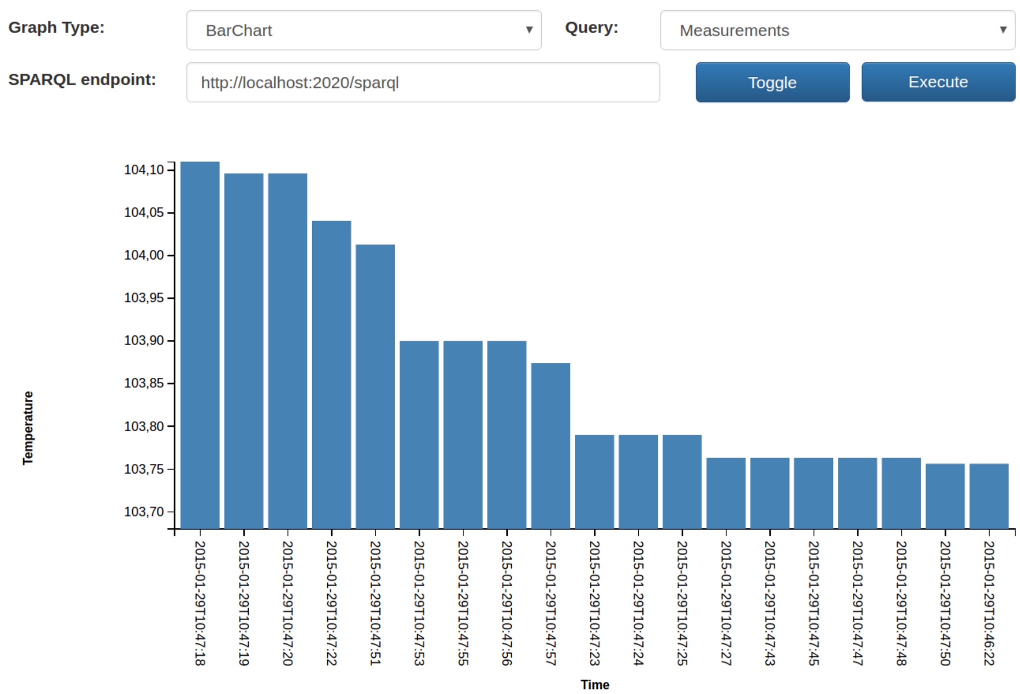


Figure 5.3: **Temperatures measured in a time interval of one hour.** The Figure illustrates values of temperature in an interval of one hour. The temperature values are obtained from the sensor AirProbe dataset described by the RAMI4.0 ontology.

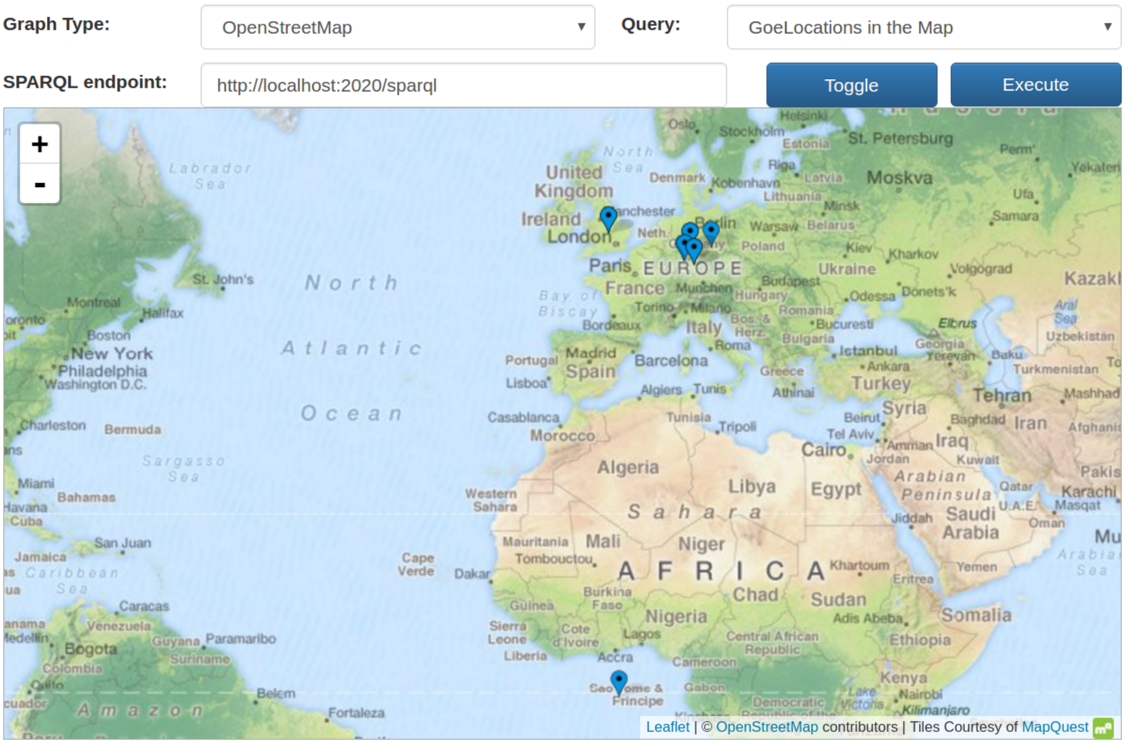


Figure 5.4: **Geolocations of the queried sensors.** Showing the results of the query in listing 5.5; geolocations of the active sensors in a time interval of one hour.

```
PREFIX rami: <https://w3id.org/i40/rami#>
PREFIX geo:  <http://www.w3.org/2003/01/geo/wgs84_pos#>
PREFIX ssn:  <http://purl.oclc.org/NET/ssnx/ssn#>
PREFIX xsd:  <http://www.w3.org/2001/XMLSchema#>

SELECT  ?assetID  ?lat  ?lng
WHERE  {
    ?adminShell  rami:describes       ?asset ;
                 rami:comprises       ?measurement .
    ?asset       rami:hasAssetId      ?assetID .
    ?measurement rami:receivedAtTime  ?time ;
                 geo:lat              ?lat ;
                 geo:long             ?lng ;
                 ssn:isProducedBy     ?sensor ;
FILTER  (xsd:dateTime(?time) >= "2015-01-29T10:00:00Z"^^xsd:dateTime)
FILTER  (xsd:dateTime(?time) <= "2015-01-29T11:00:00Z"^^xsd:dateTime)}
```

Listing 5.5: **Querying active sensors in a time interval of one hour.** Retrieving values of the latitude and longitude of a sensor in a time interval of one hour by using the RAMI4.0 ontology.

Representing Assets via the Semantic Administration Shell

The purpose of the communication between I40 components is to impact on each other in order to jointly carry out a task [206]. Integrating assets represented by I40 components require a semantic understanding of the description of the available functions and data. While integrating these assets, semantic interoperability conflicts between their data require to be conciliated. In order to understand this scenario, an example is presented. It is important to note that

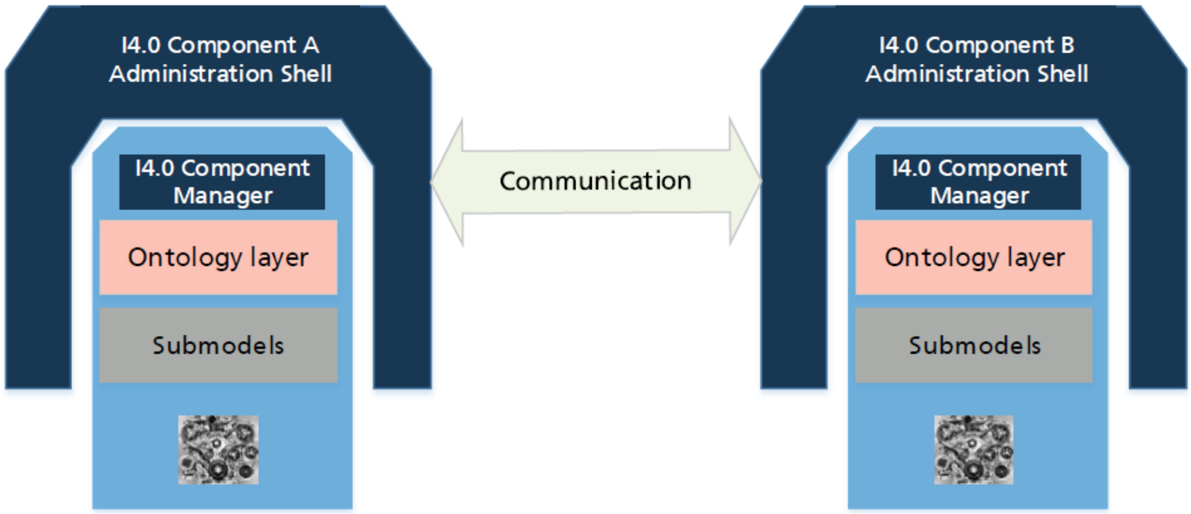


Figure 5.5: **Communication between two I40 components through its Administration Shells.** Assets, when described by their Administration Shells become I40 components. The communication between the I40 components is made through their Administration shells. An ontology layer is required to model and resolve semantic interoperability conflicts in this context (adapted from [206]).

the Administration Shell utilizes many submodels to describe the properties of an asset. In this example, we choose the submodel energy efficiency to be analyzed. The submodel energy efficiency aims to provide an example of how assets and their administration shells provide current consumption values. Figure 5.5 depicts an example of two I40 components that are competing for a manufacturing order in a marketplace scenario [203]. These two components need to communicate to respond to the comparison of energy consumption between them. The Administration Shell and the asset combined form an I40 component. The energy consumption is expressed using the energy efficiency submodel. The Administration Shell comprises the energy efficiency submodel. This scenario assumes that the energy consumption is expressed in different units of measurements, i.e., the first I40 component consumes 17.3 Kilowatts per hour (KwH) and the second 62280 Kilojoules (Kj). The values of the consumption are actually expressing the same value with different units of measurements. When representing this situation literally, i.e., by means of strings, the semantic interoperability conflicts of **SIC3** and **SIC4** appear.

The RAMI4.0 ontology permits to semantically model this scenario in order to resolve existing semantic interoperability conflicts. To accomplish this, in the following, the semantic representation of two I40 components, i.e., an asset and its administration shell are shown (cf. Figure 5.6). The RAMI4.0 ontology models these I40 components by means of their administration shells and the energy efficiency submodel. The OM ontology is utilized to express the meaning of the values for energy consumption in KhW and kJ. The `owl:sameAs` predicate is used to indicate the equivalence between the two instances. Based on this representation, the query 5.6 allows for retrieving the information regarding the two I40 components independently of the units of measure used to express the energy consumption value. Thus, resolving the existing semantic interoperability conflicts.

```
PREFIX rami: <https://w3id.org/i40/rami#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>

CONSTRUCT  {
    ?value1  owl:sameAs  ?value2.
```

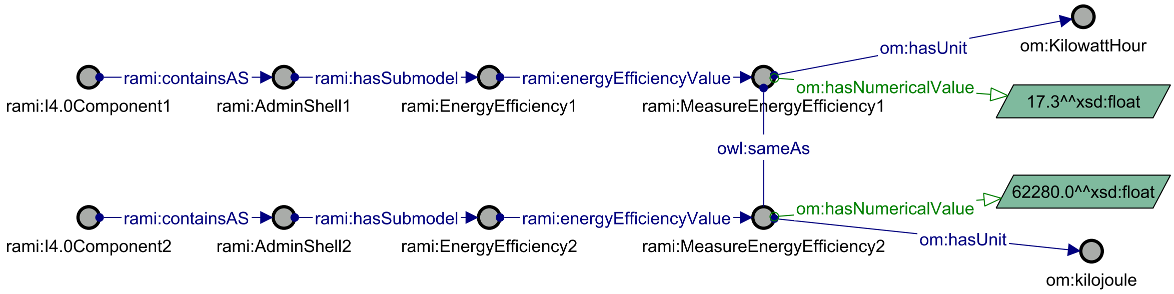


Figure 5.6: **Semantic representation of the energy consumption of two I40 components.** The RAMI4.0 ontology models these I40 components by means of their administration shells and the energy efficiency submodel. The OM ontology expresses the meaning of the energy consumption values in KhW and kJ. The `owl:sameAs` predicate is used to indicate the equivalence between the two instances.

```
    ?I40Component  rami:containsAS  ?ass.
}
WHERE  {
    ?I40Component  rami:containsAS  ?ass.
    ?aas  rami:hasSubmodel  ?submodel.
    ?submodel  rami:energyEfficiencyValue  ?value1.
    ?value1  owl:sameAs  ?value2. }
```

Listing 5.6: **SPARQL query to retrieve the I40 components with the same energy consumption value.** The energy consumption value is retrieved for the two I40 components by means of their administration shells and their energy efficiency submodels.

The presented use cases show how the Semantic Administration Shell provides a flexible data model allowing to add meaning to assets in I40 scenarios. The semantic representation of the Administration Shell enables to represent every submodel by means of a domain ontology. Then, it links the domain ontology with the main representation of the Administration Shell by relying on the RAMI4.0 ontology. Moreover, this representation also allows describing different domains of relevance for I40 scenarios, e.g., the units of measurements. By describing these domains this approach enables the resolution of semantic interoperability conflicts in I40 scenarios.

5.3 AutomationML Ontology: Modeling CPS for Industry 4.0

In this section, we describe an ontology for representing the AML standard. The AML standard is a solution for data exchange focusing on the domain of automation engineering. The emerging standard provides means for facilitating uniform data exchange between engineering tools.

An ontology-based representation of the AML standard enables the following improvements:

- flexible schema refinement and heterogeneous data linking and integration;
- using the semantic technology stack to enhance the engineering processes in CPS engineering;
- connecting to other industry standards that already have semantic representations, e.g., eCl@ss catalog [207] or the GoodRelations [208] ontology through ontology reuse or linked data mechanisms; and

- connecting to representations from other domains, e.g., eCore in Model-Driven engineering; to name just a few.

Several approaches already aimed to provide a semantic representation of AML [101–105, 209]. However, these efforts have the following shortcomings (cf. Table 3.2):

1. they do not cover the complete standard and are tailored for specific use cases;

2. they are not developed according to best practices for ontology design; and

3. they are not openly available for consultation, extension or improving current design.

The AML Ontology (AMLO) described in this section advances the state of the art by addressing the conflicts above as follows. This section is based on the following publication [171]. [17] Firstly, AMLO is an OWL ontology that covers the entire emerging data exchange standard in the CPS engineering field. Secondly, AMLO is developed using the methodology described in Section 5.1. We took as input two ontologies, developed independently of each other as initial efforts to cover the AML standard for specific applications. The first one was created using a *top-down* modeling approach [210]. The focus of this ontology was to capture the major concepts in AML, i.e., classes and relations between them, and with the goal of enabling consistency checking between different AML files. However, the property coverage is not sufficient to match the AML XSD schema specification, especially for data properties. The second ontology was developed as a part of the Alligator approach [211]. This ontology followed a *bottom-up* modeling approach and therefore had a well-elaborated property structure, but is limited w.r.t. a class hierarchy. Combining both ontologies allowed for better class and property coverage regarding the AML XSD schema and the AML standard specification. Thirdly, AMLO is openly available following best practices for ontology sharing and publishing, in particular, those described in [212]. As a result, AMLO covers 17 classes and 87 properties, which are aligned with three well-known ontologies, i.e., the *PROV* ontology, the *OM* ontology, and the *skos* vocabulary.

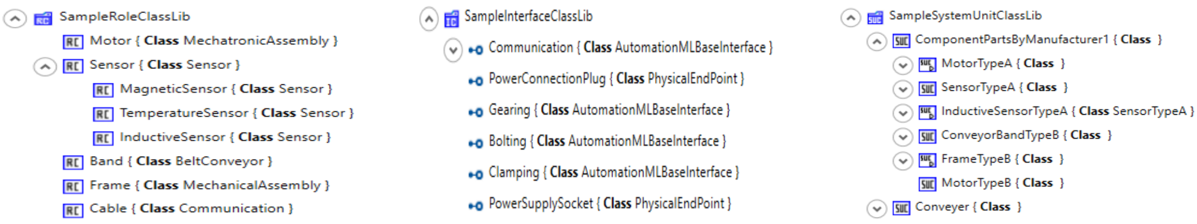
5.3.1 Context. AutomationML and Engineering Design

AMLO is mainly focused on modeling topology information by means of the CAEX standard, which is a core part of the AML standard. The following CAEX concepts are the basis of AML.

RoleClassLibrary contains a collection of possible functionalities that can be provided by the plant equipment. A *Role Class* (RC) defines a physical or a logical object as an abstraction of a concrete technical realization, which is vendor independent, e.g., a robot or a sensor. This way a functional semantics is assigned to an object, enabling an automatic interpretation by a tool. RCs can also define attributes that are generally expected for this object type, e.g., a payload for a robot RC. Additionally, RCs can be assigned to objects within the SystemUnitClassLibrary and InstanceHierarchy to specify an object type.

InterfaceClassLibrary defines all interfaces required to describe the plant model. An *interface class* (IC) can be used either: a) for specifying relations between objects of a plant topology, e.g., to connect a sensor with a PLC; or b) for specifying references to information that is stored outside of the CAEX file, e.g., to assign a 3D description to an object.

[17] This is a joint work with Olga Kovalenko, a PhD student at the Technical University of Vienna. My contributions to this article comprise the ontology modeling and implementation, contributions to an overall problem definition, use cases, as well as the analysis and revision of the related work.



(a) RoleClassLibrary (b) SystemUnitClassLibrary (c) InterfaceClassLibrary

Figure 5.7: **AutomationML modeling example**. A Conveyor consisting of a frame, a band, an inductive sensor, and a motor. (a) SampleRoleClassLib contains role classes that define vendor independent functionalities for the conveyor. (b) SampleSystemUnitClassLib lists vendor specific device realizations options for the conveyor. (c) SampleInterfaceClassLib comprises Interface classes for the conveyor.

SystemUnitClassLibrary comprises collections of vendor specific solution equipment objects. Those objects can be matched with the system requirements, defined by the role classes, and used to implement the plant design. A *System Unit Class* (SUC) defines a concrete technical realization of a physical or logical object, thus representing a specific instantiation for an RC. System unit classes are instantiated within the InstanceHierachy.

InstanceHierarchy describes the plant topology, including the concrete equipment of an actual project, i.e., the instance data. The instance hierarchy contains all data including properties, interfaces, role classes, relations, as well as references.

Applying AML in engineering projects is already an important improvement, as it facilitates data exchange between the project stakeholders and defines a project-specific vocabulary to which all engineers can relate. Nevertheless, there is still a lack of infrastructure for supporting advanced engineering activities across the various engineering disciplines and their corresponding tools, e.g., data linking, change propagation across connected datasets, data analysis, and consistency checking. Having an ontology for AML help to address these gaps.

An AutomationML Modeling Example

To illustrate how a CPS is represented in AML, we developed an example using the AML Editor 4.7.0.[18] The AML Editor allows for browsing and developing AML files. The example system models a conveyor that consists of a frame, a band, an inductive sensor, and a motor.

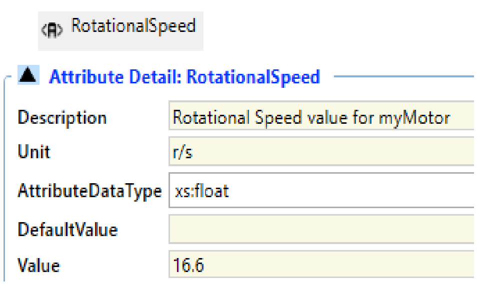
The RoleClassLibrary contains classes to represent the basic functionalities for the system components, in our case *Motor*, *InductiveSensor*, *Band* and *Frame*. The *Frame* and *Motor* extend the basic class *MechatronicAssembly* that is defined within *AutomationMLBaseRoleClassLib*. Different types of sensors are represented by the RCs *MagneticSensor* and *InductiveSensor*, all of which are derived from the more general *Sensor* RC (cf. Figure 5.7(a)). The role classes are vendor independent and do not comprise implementation-specific details, e.g., a role class *Motor* means "something that can fulfill motor function".

The InterfaceClassLibrary contains interface classes representing various relations between the system components. These relations can be of different nature: mechanical ones, e.g., *Bolting* and *Gearing*; electrical, e.g., *PowerConnectionPlug* and *PowerSupplySocket*; of software and control related, e.g., *Communication* (cf. Figure 5.7(b)).

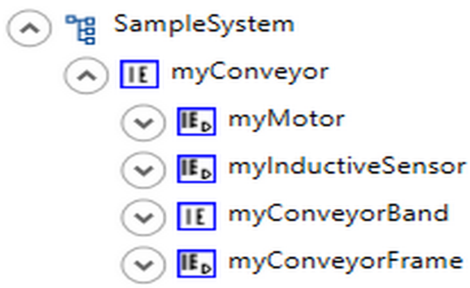
The SystemUnitClassLibrary comprises component realizations that can be used to implement

[18] https://www.automationml.org/o.red.c/dateien.html

the functionalities, defined by role classes in a real system. System unit classes are vendor specific, e.g., *MotorTypeA* represents a specific device with the motor functionality produced by Manufacturer1 (cf. Figure 5.7(c)). Role classes are assigned to SUCs to specify the functionalities that a specific device can fulfill. E.g., *SensorTypeA* has the RC *Sensor* assigned, meaning that it can be used to implement the functionality defined by *Sensor* RC. Additionally, attributes are specified for the internal elements providing further details for the specific devices. For example, *MyMotor* has the attributes *Weight*, *Material*, and *RotationalSpeed* defined. For each attribute, a description, a measurement unit, a data type, and value can be specified. For instance, one can see that the *RotationalSpeed* attribute comprises the description, the value of 16.6, the unit "r/s", and the `xs:float` as data type (cf. Figure 5.8(a)).



(a) RotationalSpeed Attribute of MyMotor



(b) Hierarchy classes for the sample system

Figure 5.8: **Description of attributes of a simple AML hierarchy.** The RotationalSpeed attributes of the Motor (MyMotor) in the sample system are shown, i.e., Description, Unit, AttributeDataType, and the actual value. Levels of the class hierarchy of the sample system are described, i.e., from the myConveyor, myMotor, and BoltingFrame class. The AML standard enables the hierarchical representation of the elements of a CPS. The properties and the values of the elements can be also described.

Finally, the InstanceHierarchy contains the design of a real-life system, i.e., the conveyor consisting of a band *myConveyorBand*, a motor *myMotor* to move the band, a frame *myConveyorFrame*, and an inductive sensor *myInductiveSensor* to identify positioning of the object on the band (cf. Figure 5.8(b)).

5.3.2 Methodology

In this section, we describe the modeling process for the AML Ontology (AMLO) by applying the proposed methodology to build ontologies of I40 related standards. The specific design decisions to cover the AML standard are also reported, e.g., an overview of the ontology structure with major classes and properties. Next, ontologies and ODPs that are reused in AMLO are discussed and best practices are observed.

5.3.3 Purpose and Scope

The design of AMLO follows the following points:

- AMLO is required to be maximally compatible with the AML XSD schema and the standard;

- The vocabulary used by AMLO is required to be as close as possible to the terminology in the standard, to allow an intuitive understanding of the ontology by users from industry who are familiar with the standard, but not with semantic technologies.

5.3.4 Capture

As a starting point for ontology development, we relied on two ontologies, initially designed with the intention to provide semantic representation for the AML standard for specific applications. The first ontology was built at TU Vienna using top-down modeling approach [209]. The result is a light-weight ontology capturing the major design decisions in terms of classes and relations between AML elements. The intended application is consistency checking between different AML documents in multidisciplinary engineering projects. However, property coverage is weakly elaborated and do not completely cover the AML XSD schema specification, particularly for data properties. The second ontology was built following the bottom-up modeling approach as part of the Alligator project at the University of Bonn [211]. The intended application is the automatic semantic integration of AML documents. This ontology had a well-elaborated data property structure, but a less developed hierarchy for classes and relations. The conceptualization phase of AMLO is therefore accomplished by combining both ontologies and refining the conflicting concepts, where necessary. This allowed for better class and property coverage w.r.t. the AML XSD schema and the AML standard specification. While working on the AMLO especial emphasis is put on following the best practices for ontology design [213, 214].

With the goal of validating the ontology resulting from the previous steps, we applied the following approach. First, we performed two iterations of structure validation with domain experts. For this, we asked the support of colleagues in the Otto-von-Guericke University (Magdeburg), who are actively involved in the development of the AML standard and are, therefore, deeply familiar with the semantics of the AML constructs. Second, we manually implemented several AML data samples, provided on the official AML web-site[19] by means of the AMLO. This guaranteed that various modeling situations supported by the standard can be indeed described by means of AMLO.

5.3.5 Design

Flexibility of the standard To allow for a wide adoption among industrial practitioners, the standard creators intentionally avoided including too many constraints in the AML standard specification level. For instance, the general design approach in AML assumes that one first defines the functional capabilities for a future system using role classes (RCs) and then one selects and assigns suitable system unit classes (SUCs), which is a concrete realization of those capabilities defined via RCs. However, it is also possible and is not an error if one defines a system directly using the SUC libraries. Hence, the tool vendors that typically have elaborated SUC structures, but only limited (if any) hierarchies of general functionalities, could use AML without having to significantly adjust their workflows. With the aim to support the flexibility of the AML standard in the designed ontology, we decided to keep the number of predefined restrictions to a minimum. This way, modeling with the AMLO have the same flexibility as of when using the AML standard.

Modeling hierarchies of ICs, RCs and SUCs There are three major options to model the hierarchies of interface classes (ICs), RCs and SUCs in AMLO. First, one can build

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

class hierarchies to capture hierarchical relations between the elements of IC-, RC- and SUC-libraries in AML. For example, `aml:MagneticSensor` and `aml:InductiveSensor` are both subclasses of `aml:Sensor`, which is a subclass of a `aml:RoleClass`. Instances of `aml:InductiveSensor` describe specific roles assigned to a certain device. These are represented via instance of an `aml:InternalElement` class. For instance, `aml:InductiveSensor` role assigned to a specific device `aml:myInductiveSensor` in the instance hierarchy (see modeling example in Section 5.3.1). Secondly, all roles can be modeled as instances of the RC concept. In this case, the further hierarchical relations between the individual roles are modeled via additional constructs in the ontology, which relate the RC instances to each other. The third option consists in capturing different roles via relations, e.g., `aml:hasSensor` that connect instances of `aml:InternalElement` corresponding to specific devices with instances of `aml:RoleClass`. The hierarchical relations between the roles are then modeled using the subproperty mechanism, e.g., `aml:hasMagneticSensor` and `aml:hasCurrentSensor` being subproperties of `aml:hasSensor`. These three possibilities for modeling the same semantics are compatible, i.e., rules can be defined to automatically transform one into another [183]. Inspired by the experiences in studying well-known ontologies, the first approach is adopted for modeling the IC, RC, and SUC hierarchical relations in AMLO. This approach seems to be the most intuitive in terms of similarity to the AML representation.

Splitting up multiple-use properties with `rdfs:subPropertyOf` There are some relations in AML that capture the same semantics, but for different object types. For example, an AML file can contain all four AML main object collections, i.e., InstanceHierarchy, InterfaceClassLib, RoleClassLib, and SystemUnitClassLib. In this case, the `aml:contains` property holds the same semantics for all four constructs. Therefore, the straight-forward way of modeling this scenario is to simply define `aml:contains` object property that relate instances of a `aml:CAEXFile` to the instances of `aml:InstanceHierarchy`, `aml:InterfaceClassLib`, `aml:RoleClassLib`, and `aml:SystemUnitClassLib`. However, this can potentially cause problems during reasoning and inference tasks, because of the Open Word Assumption logic implied by reasoners. To tackle this problem, we decided to model such relations in the following way: a) a super property is defined capturing the general relation semantics. This super property does not explicitly specify a certain class for its range. For instance, the domain of the `aml:contains` property is the `aml:CAEXFile`, but nothing is specified for the range; b) subproperties are defined to describe each specific case for the property range, e.g., `aml:hasInstanceHierarchy`, `aml:hasInterfaceClassLib`, `aml:hasRoleClassLib`, and `aml:hasSystemUnitClassLib` are defined to be subproperties of `aml:contains`. This approach allows using a less complex vocabulary for the applications where only querying is important. That is, one can define relations using the general `aml:contains` property without having to learn the property name for each specific case. At the same time, if the intended application involves reasoning one can use more elaborated and detailed vocabulary, e.g., subproperty labels, to avoid potential conflicts.

Linking to eCl@ss and domain-specific standards An important feature of AML is the ability to link to eCl@ss – the standardized catalog for describing products and services. In AML, the semantics of some of its elements, e.g., RCs and attributes, can be specified by linking them to corresponding eCl@ss definitions. Namely, attributes are semantically equivalent if they share the same value for the `refSemantic` attribute; and RCs are semantically equivalent whenever they share the same eCl@ss references for AML attributes such as `eClassVersion`, `eClassClassification`, and `eClassIRDI` [215].

AMLO also supports linking to the eCl@ss standard via the class `aml:ExternalStandard` and the object property `aml:hasRefSemantic`, which has the class `aml:ExternalStandard` as its range.

Following the strategy explained above, two subproperties of the `aml:hasRefSemantic` are further defined, with `aml:RoleClass` and `aml:Attribute` as a domain and `aml:ExternalStandard` as a range. Thanks to the general nature of this linking mechanism, one can also link to other domain-specific standards. To this end, a corresponding instance of the class `aml:ExternalStandard` is created. The linking is then done in a similar way as described above for the eCl@ss standard.

Ontology Overview

By considering the aforementioned design decisions, AMLO is modeled with a focus on the main AML elements. AMLO covers all the major constructs of the CAEX XSD schema, with one exception for the family types for RCs, SUCs and ICs, i.e., *RoleFamilyType*, *SystemUnitFamilyType* and *InterfaceFamilyType*. Those are XML related structural concepts that are needed in XML to specify the parent-child hierarchical relations. Since OWL provides means to model such relations explicitly there is no need to keep those construct in the ontology. Figure 5.9 shows the main entities of the ontology.
aml:CAEXFile This class represents the AML document and is the core element of the ontology. The class `aml:AdditionalInformation` is connected to the `aml:CAEXFile` class to describe the metadata related to the document. It comprises data about the version, writer identification, name, release, vendor as well as the project to which the document belongs.
aml:RoleClass represents the role class concept in AML. This concept defines a vendor independent functionality that can be provided by equipment elements. The class `aml:RoleClass` is used to assign generic functional semantics to instances of the classes `aml:InternalElement` and `aml:SystemUnitClass`. The objective is to describe the functional capabilities of equipment elements. This is done via the properties `aml:hasSupportedRoleClass` and `aml:hasRoleRequirements`.
aml:SystemUnitClass represents the AML *system unit class* concept and defines a vendor-specific technical realization of a physical or logical object. Instances of the `aml:SystemUnitClass` can contain and be a part of other system unit classes. This is defined via properties `dul:hasPart` and `dul:isPartOf` respectively.
aml:InterfaceClass represents the interface class concept in the AML and in general is used to specify either a) relations between the different topology elements; or b) references to various external information sources. Interfaces can be linked to the instances of `aml:RoleClass`, `aml:SystemUnitClass`, and `aml:InternalElement` via the `aml:hasInterface` property. The class `aml:ExternalInterface` is modeled as a subclass of `aml:InterfaceClass`.
aml:InternalElement models the concept of internal elements in AML and defines concrete equipment of an actual project, i.e., devices used in a concrete plant. The class `aml:InternalElement` can contain, and be part of other `aml:InternalElement` classes. This is defined via properties `dul:hasPart` and `dul:isPartOf`. One can additionally define what `aml:SystemUnitClass` is instantiated by a specific `aml:InternalElement` via the `aml:hasBaseSystemUnitClass` property.
aml:Attribute expresses the concept of an AML attribute. An AML attribute describes actual properties of a CPS, e.g., length, size, temperature, speed. Attributes can be connected with the instances of `aml:RoleClass` and `aml:InterfaceClass` via the `aml:hasAttribute` property. The class `om:Measure` is reused from the OM Ontology to define the units of measure for attributes.
aml:InternalLink is an interesting element of AML that represents a directed connection between two constructs. Each `aml:InternalLink` references two `aml:ExternalInterface` and two constructional elements, i.e., either `aml:InternalElement` or `aml:SystemUnitClass`, depending on which level of abstraction of a given connection is specified. This is done via properties `aml:hasRefPartner(A/B)Interface` to link to the class `aml:ExternalInterface`, and

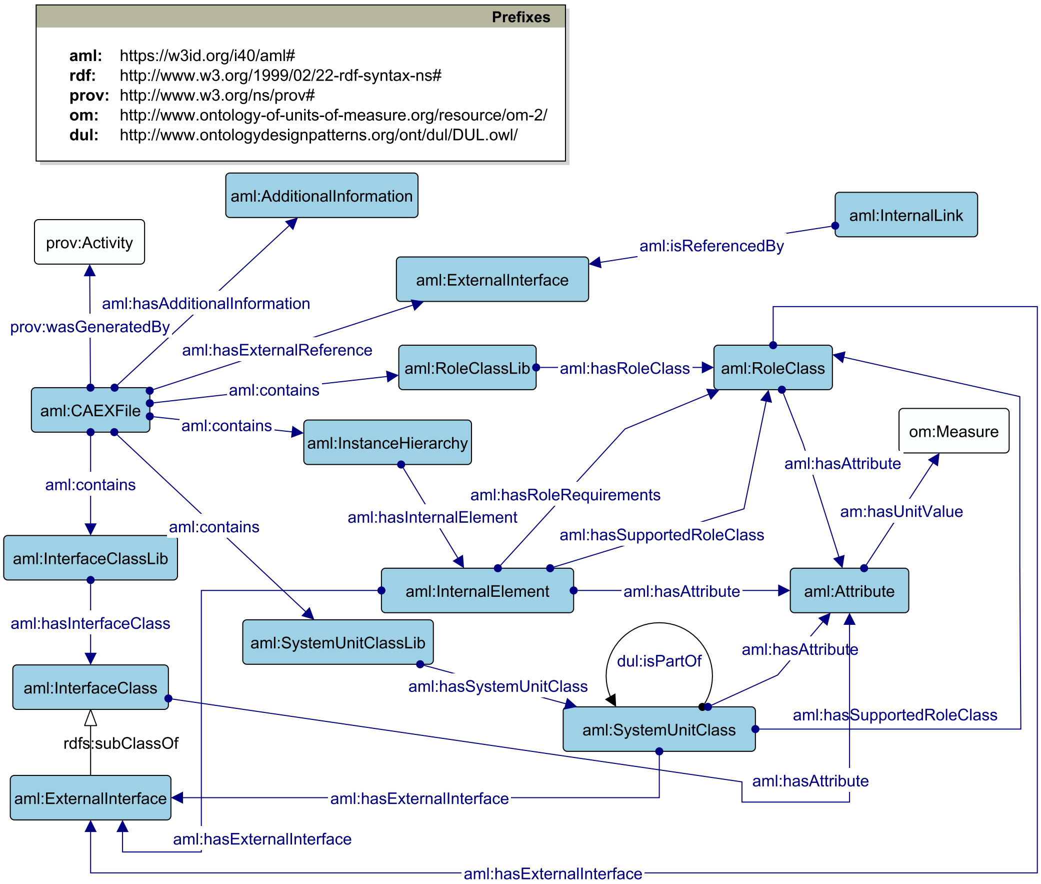


Figure 5.9: **Core classes and properties of AMLO.** Core AML concepts are represented in AMLO by means of classes and relations among them. Classes and relations of AMLO are depicted in blue. White classes represent reused classes from well-known ontologies such as PROV and OM.

via properties `aml:hasRefPartner(A/B)Object` to point to the classes `aml:InternalElement` and `aml:SystemUnitClass`. The direction is important in this case, i.e., "from A to B".

Reusing Well-known Ontologies

Examples of reusing are ontologies that cover engineering and related topics, or domain-specific sources, e.g., product catalogs like eCl@ss [207]. Once having the semantic representation of the original AML data, existing semantic data sources can be used to enrich original data structures with additional information. Even though AML comprises very specific concepts and relations, we reused well-known ontologies such as the *skos* vocabulary, the *PROV* Ontology, as well as the *OM* Ontology. The *skos* vocabulary is employed to encode additional information about concepts and property related to documenting, e.g., definitions for AMLO constructs or alternative labels. The *PROV* ontology is used to represent the information of the activities that generated the AML documents by means of the class `prov:Activity`. The utilization of the OM ontology is explained in Section 5.3.5.

Ontology Design Patterns

Hereby, we discuss existing ODPs and how they are applied to support the modeling of the AMLO. In AMLO the transitive version of the `partOf` ODP is used. We selected this option for part-whole relation because in the context of engineering system design a potentially important application is recursively getting all parts and subparts of a specific object. The *Componency ODP* or the *TimeIndexedPartOf ODP* are not used for a specific time interval in AMLO, since there are no relations with similar semantics in the AML standard. Another pattern, which is often discussed in the context of part-whole relations, and is also relevant in the AML context is the *Constituency ODP*. In the context of AML, the constituency is typically represented by an attribute, i.e., `aml:hasMaterial` or similar defined for SUCs or IEs. Since the attribute hierarchy can vary from one AML file to another, we decided to model this aspect as follows: a) an object property `aml:hasConstituent` is included in the ontology; b) after a specific AML file is transformed into the AMLO, and, therefore, a certain property hierarchy is formed under the `aml:hasAttribute` property, one can align the specific attribute, e.g., `aml:hasMaterial` with the `aml:hasConstituent`, specifying that these two have similar semantics.

Connections system components represent interactions between its parts, such as flows of energy, matter or signals [216]. In AML, connections are modeled via ICs. ICs allow specifying connection elements which establish the connection, direction, and connection type. For instance, a signal transfer for control variables or different kinds of mechanical linking between the system components. ICs allow for a detailed and flexible definition of various connections in an engineering system. Therefore, we decided not to use additional ODPs in the ontology for modeling connections.

Component roles depict the functionality and behavior associated with a component in a system [86]. In AML, such information is represented via RCs. An RC is specified for an IE or SUC to define what kind of functionality in a system this component has. Hence, we did not use any additional ODPs to model this kind of relations, since they can be modeled directly by means of AML constructs that AMLO derives from the standard.

Units of Measurements

In I40 scenarios, units of measurement are of key relevance for the correct function and coordination of the associated processes. The *OM* ontology is used to bring formal semantics for encoding units or measurement, which are originally represented as strings in AML. Since the problem of disambiguation of units of measure is a very common and relevant for the engineering domain, we focus on it in more detail. Below we detail the use of OM and how it can be exploited in the AMLO to bring more semantics into the units of measure.

```
<Attribute Name="RotationalSpeed" Unit="r/s" AttributeDataType="xs:float">
    <Value>16.6</Value>
</Attribute>
```

Listing 5.7: **RotationalSpeed attribute of the motor myMotor in AML.** A fragment of the attribute RotationalSpeed encoded in the AML standard; The Unit, AttributeDataType, as well as the Value are shown.

Unit of measurements are required for the specification of products and for representing the data produced by measuring devices. Commonly, data related to units of measurement are codified as simple strings, thus, losing the semantics associated with them. Additionally, this way of representing the units of measurement causes semantic interoperability conflicts to arise.

```
@prefix aml: <https://w3id.org/i40/aml#> .
@prefix om : <http://www.ontology-of-units-of-measure.org/resource/om-2/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

aml:SpeedAttribute        a  aml:Attribute;
    aml:hasNameAttribute  "Rotational Speed"@en;
    aml:hasMeasure        aml:spMeasurement.

aml:spMeasurement         a  om:Measure ;
    om:hasUnit            om:radianPerSecond-Time;
    om:hasNumericalValue  "16.6"^^xsd:float.
```

Listing 5.8: **Representing units using the OM ontology.** Representing the unit of measure of the RotationalSpeed attribute, i.e., radians per second by combining AML and OM ontologies.

In AML, attributes are used to express properties of different objects. For instance, consider the motor from the sample conveyor system presented in Section 5.3.1. An important attribute of a motor is a rotational speed, which can be measured both in radians per second and in revolutions per second. If represented informally as a string, both of those units can be expressed as “r/s”. Listing 5.7 depicts a fragment of an AML document showing an attribute setting the rotational speed of the motor. The intended meaning of “r/s” in this case is radians per second but could be also interpreted as revolutions per second. This example demonstrates the importance of semantically representing units of measurement to avoid ambiguity, as well as to express the correct semantics of the attribute. In this case, semantic interoperability conflicts of **SIC3** and **SIC4** exist. As a solution, Listing 5.8 represents the example, semantically combining the AMLO and the OM ontologies.

5.3.6 Summary of Ontology Characteristics

For the design of AMLO, best practices of ontology development presented in Section 5.1.1 are used. Below, specific practices that are employed for AMLO are described. The documentation of the ontology is publicly available via a *VoCol* instance.[20] The major goal of the ontology is to fully cover the AML standard while preserving its flexibility. Therefore, AMLO can be used in a wide range of scenarios. AMLO can be aligned to other standards using the `aml:ExternalStandard` class and the object property `aml:hasRefSemantic`. AMLO has the Creative Commons license and is available online.[21] The source code of the ontology as well as the evolution track are also publicly available on GitHub.[22] The reuse of well-known ontologies and the consideration of ODPs are included in the design process of AMLO. With the objective to ensure that AMLO is self-explanatory, we included the following skos constructs: a) `skos:definition` to explain the meaning of main entities; b) `skos:altLabel` to include the alternative names for ontology entities, e.g., “SUC” for “System Unit Class”; and c) `skos:prefLabel` to describe the most commonly used name for ontology entities. Table 5.4 summarizes the main characteristics of AMLO.

5.3.7 Use Cases

In this section, two representative use cases supported by AMLO are presented. The use cases show how the querying (cf. Section 5.3.7) and reasoning (cf. Section 5.3.7) can be applied on top of integrated data. The context for the two use cases is the develop of an automation

[20] http://vocol.iais.fraunhofer.de/aml/
[21] https://w3id.org/i40/aml#
[22] https://github.com/i40-Tools/AutomationMLOntology

Table 5.4: **Summary of the AML Ontology Characteristics.** The table shows a summary of AMLO in aspects such as general details, reused ontologies, documentation, naming conventions, multilinguality, availability, and used methodology.

General	**Name:**	AutomationML Ontology (aml)
	Size	21 classes, 51 object properties, 46 data properties
	DL Expressivity	ALHIF+(D)
Reuse	**P-R1**, **Reused Ontologies**	PROV, OM, SSN, DUL, FOAF
	P-R2, **Reused ODPs**	transitive PartOf ODP
Documentation	**P-Do1**, **P-Do3**	By means of `skos:definition`, `skos:altLabel`, `skos:prefLabel`
Naming Conventions	**P-N1 - P-N8**	CamelCase notation
Multilinguality	**P-M1**, **PM2**	English for all terms and german labels for core elements
Availability	**PersistentURI**	`https://w3id.org/i40/aml#`
	GitHub	`https://github.com/i40-Tools/AutomationMLOntology`
	LOV	`http://lov.okfn.org/dataset/lov/vocabs/rami`
	Licence	Creative Commons
	VoCol Instance	`http://vocol.iais.fraunhofer.de/aml/`
Methodology	**Ont. Eng. Methodology**	King and Uschold [176]

system, e.g., a production plant and corresponding control system. Such engineering setting is characterized by a rich variety of engineering tools, terminologies, heterogeneous data models, and data formats applied by project stakeholders. Stakeholders work on the same engineered system, but consider it from different points of view, e.g., plant planning, mechanical engineering, electrical wiring or control system implementation. Therefore, this is a complex and data-rich setting, with high demand for effective and efficient data integration and advanced analytic within and across the disciplines. Project participants apply AML as a common vocabulary for data exchange, which is a first step to facilitate data exchange and communication. However, there is still a need for tool-support and technologies, which allow: a) data integration across the different engineering disciplines and b) comprehensive and flexible analytic over the system's engineering data. Semantic technologies possess rich capabilities for both tasks, e.g., by offering comprehensive querying with SPARQL as well as inference and reasoning facilities.

Weight and Power Consumption of a Production Model

After the AML data from various disciplines is integrated, e.g., SPARQL queries can be used to perform analysis on the top of engineering data. Hereinafter, we present the example consistency check that is implemented in the use case of our industry partner, a power plant system integrator, based on AMLO. In the example, we used AMLO to access the device characteristics for gathering the overall system statistics. It is assumed there are maximum weight and electrical consumption thresholds specified by a customer for the system under design. A project engineer can run the following SPARQL query over the engineering data to obtain this information.

```
PREFIX aml:  <https://w3id.org/i40/aml#>.
PREFIX xsd:  <http://www.w3.org/2001/XMLSchema#> .
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

SELECT  (SUM (xsd:integer (?deviceWeight))  AS ?systemWeight)
        (SUM (xsd:integer (?devicePowerConsumption))  AS ?systemPowerConsumption)
WHERE  {
    aml:myConveyor  aml:hasPart*  ?device
    ?device         a  aml:InternalElement .
    ?device         aml:hasAttribute  ?attribute .
    ?attribute      aml:hasAttributeName  "Weight"@en .
    ?attribute      aml:hasValue  ?deviceWeight .
    ?device         aml:hasAttribute  ?attribute .
    ?attribute      aml:hasName  "PowerConsumption"@en.
```

```
    ?attribute      aml:hasValue  ?devicePowerConsumption . }
```

Listing 5.9: **Returning the weight and power consumption.** SPARQL query returning the overall weight and power consumption of a production model by using data described by AMLO.

Flexible Hierarchy Adaptation using Reasoning

For this use case, we assume there is the following requirement defined by a project engineer: "All controller devices in a production system must have exactly one connection to an automation object". However, the "controller" role is not defined explicitly in the project and must be defined separately for each topology of roles. The ontology-based representation of AML data allows flexible reconfiguration of the defined structures. Reasoning can be applied to the roles hierarchy in order to enrich the existing classification, e.g., the following SWRL rule can be defined to automatically classify the controllers. In this case we are assuming that the marker for being a controller is supporting the "RoMechatronicAssembly" role.

```
SystemUnitClass(?device_type) ∧ RoleClass(?role) ∧ hasSupportedRole(?device_type, ?role) ∧ RoMechatronicAssembly →
Controller(?device)
```

Listing 5.10: **SWRL rule for reclassification of the RoleClass hierarchy.** The rule relies of the semantics encoded in AMLO for reclassifying the RoleClass hierarchy.

This rule enriches the existing roles hierarchy. The newly derived triples, i.e., knowledge, can be then materialize into the ontology. In this way, the new classification is available for all later check executions. All devices in the production system can be now automatically checked whether they are controllers or not. After this classification is performed, one can run a SPARQL query to check whether all controller devices in the system have the required property.

5.4 SCORVoc: Ontology-based Information Integration and Exchange in Supply Networks

In the past decades, internal enterprise information systems experienced much technical and scientific advancement. However, comparatively little progress is made to improve the exchange of information between factories. Until today, most of the communication between factories is done via informal channels, such as emails or telephone calls. Only tier-1 suppliers of major Original Equipment Manufacturers (OEM) are usually fully integrated into the information exchange and corresponding IT-support, e.g., electronic data interchange connections, since these are expensive to deploy and maintain. Informal communication is time-consuming, costly and can become inefficient when crucial information is spread among different people that use their own format or system. The horizontal integration needed in I40 scenarios is based on the capability to semantically describe data among practitioners in the production chain.

This section focuses on the two following business requirements:

1. The production plans of a factory are highly dependent on the incoming supplies since just-in-time production aims at keeping the stock as low as possible to reduce *dead capital*. Hence being able to instantly exchange machine-interpretable messages between manufacturer and supplier is critical, for example, in case of supply shortages.

2. It is a competitive advantage for a business to be able to assess the reliability of suppliers. However, monitoring the direct suppliers is not enough, since problems deeper in the

Supply Chain (delays, strikes, outages, bankruptcies) can have a negative effect even on reliable suppliers. Therefore, it is of paramount importance to be able to pro-actively identify critical suppliers and potential threats in the entire value-added network.

Hence, a standardized way of representing knowledge about the business processes and the supply network is needed. This section focuses on the *Supply Chain Operations Reference Model (SCOR)* [30]. While SCOR offers a basis to answer the two identified business requirements, its reference itself only contains textual definitions. Neither machine-interpretable formats for messages nor automatized ways to reason with such messages exist.

In order to address these limitations, we developed an approach for making the SCOR reference model executable. The approach presented in this section is based on the following publication [172][23]. Our approach comprises the definition of the SCORVoc vocabulary providing an ontological formalization of the terms and concepts defined by SCOR. We argue that using an ontology-based approach is a step towards applying the SCOR reference in real-world applications. Additionally, the semantics of data involved in the complete Supply Chain can be described and, thus, machine-readable. The ontological representation formalism provides a number of benefits for implementing SCOR, namely:

- *Identification* – world-wide unique identifiers facilitate the data exchange and linking in supply networks;
- *Coherence/Reuse* – mixing and mashing of vocabularies as well as schemata enables the reuse and alignment with domain-specific formalisms;
- *Granularity* – integration of representation on different levels of granularity;
- *Execution* – query execution in order to automatically aggregate and analyze data; and
- *Integration* – semantic definitions of the terms defined in the standard which allow for resolving semantic interoperability conflicts in the domain.

5.4.1 Context

The motivation behind SCOR is to enable factories to diagnose and manage their Supply Chains. Figure 5.10 illustrates the limited view of an enterprise without any Supply Chain communication. The goal is to extend the view in order to identify poorly performing links and act upon them. Besides communication, it is necessary that each link is equally measured by each partner.

5.4.2 Methodology

This section describes the application of the proposed methodology for formalizing SCOR in a comprehensive vocabulary, called SCORVoc. The main issues in existing formalizations of SCOR is that they are not developed by following best practices for ontology design, e.g., following a specific methodology for ontology building. Furthermore, they do not consider the utilization of

[23] This article is a joint work with Niklas Petersen, a PhD student at the University of Bonn. My contributions to this article are dedicated to the problem definition, ontology modeling, and the analysis and revision of the related work.

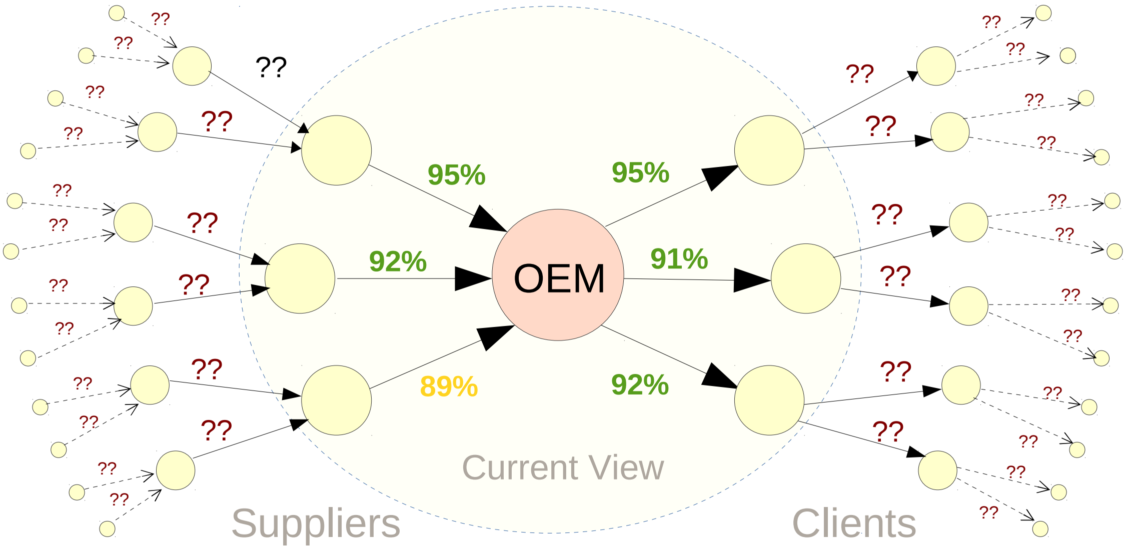


Figure 5.10: **Supply Chain workflow example: Reliability between factoriers.** Without a formalization and semantic description of the communication in the Supply Chain, a factory possess a limited view of their customers and suppliers thus, negatively affecting the horizontal integration.

ODP and are not available for continuous checking and improvement (cf. Table 3.3). On the contrary, SCORVoc is developed by using the presented methodology. Particular attention is paid to follow best practices as part of the methodology. Further, the development of synthetic data and metrics enables the usability and improvement of the vocabulary.

5.4.3 Purpose and Scope

The purpose of SCORVoc is to provide factories with a vocabulary which they can use to express any data related to Supply Chain Management (SCM). The users of the vocabulary are therefore factories which profit from the benefits of expressing their supply chains in SCOR. The vocabulary aims at being light-weight in order to facilitate its use for future SCOR compliant IT applications.

5.4.4 Capture

The capture of the domain of interest, i.e., Supply Chain data management, is achieved in two ways. First, we used the SCOR reference document that comprises 976 pages [30]. This document contains terminological definitions which represent a major source for studying the domain of interest. Second, a domain expert with deep knowledge in SCM supported the process of capturing the knowledge of the domain. The domain expert is a member of the *APICS Supply Chain Council.*[24]

[24] http://www.apics.org/sites/apics-supply-chain-council

5.4.5 Design

This step is decomposed in the design of the two main aspects of SCOR: processes, described in Section 5.4.5; and metrics, described in Section 5.4.5. Existing semantics for each concept and the lack of accessibility of prior approaches to formalizing SCOR influenced the creation of many concepts by ourselves.

Formalizing SCOR Processes

In SCOR, there are 201 processes. A process represents any business activity between and within factories. For most of them, the reference outlines unambiguous text definitions. Since some of them have a rather long name, e.g., *Identify, Prioritize And Aggregate Supply Chain Requirements*, we decided to keep the short name and attach the long version as a label. To stay coherent, all concept and property names followed the camel case notation.

As proposed in the reference, the processes are created following a hierarchical structure. An abstract super class is defined, i.e., *Process* with its subclasses *Plan*, *Source*, etc. While certain terms, e.g., *Make*, *Deliver*, do not seem to be self-explanatory, they are nevertheless adopted due to the clear meaning in the SCM domain. Altogether, the hierarchy consists of three levels, i.e., Scope, Configuration, and Step. Each level fulfills a certain purpose:

- Level 1 groups processes together,
- Level 2 comprises events, that are to be instantiated in the real world, and
- Level 3 explains in detail how level 2 processes are to be executed (step by step).

Furthermore, the reference defines IDs and clear text definitions for each process. These are reused as is in SCORVoc. Figure 5.11 visualizes the general structure of the vocabulary with the processes and others main concepts.

Listing 5.11 shows an example of the full definition of the concept *Enable*.

```
@prefix scor: <http://purl.org/eis/vocab/scor#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .

scor:Enable          a  owl:Class ;
    rdfs:subClassOf  scor:Process ;
    rdfs:label       "Enable"@en ,
                     "Erlauben"@de ;
    rdfs:comment     "Enable describes the ..."@en;
    skos:notation    "E" .
```

Listing 5.11: **Concept definition example in SCORVoc.** Example of the definition of the concept `scor:Enable` as a process. Multilinguality annotations according to the proposed methodology are added in English and German.

Enable is a subclass of the abstract concept `scor:Process`. Each concept contains a definition together with further descriptions provided by SCOR. We further added translations for a variety of languages to support Multilinguality (cf. **P-M1** and **P-M2**).

SCOR further defines 179 *Practices* to provide a collection of industry-agnostic practices which are recognized for their value. In order to manage talent in the supply chain, concepts such as *Person*, *Skills*, *Experiences*, *Aptitudes*, and *Trainings* are considered.

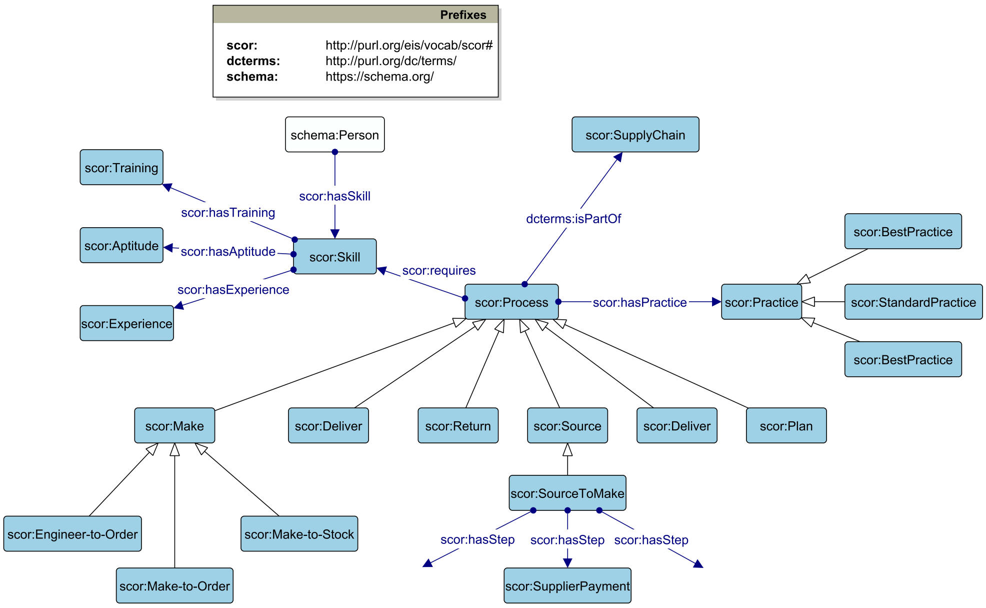


Figure 5.11: **Core classes and properties of the SCORVoc vocabulary.** Main classes and some relations are shown of the SCORVoc vocabulary. Blue squares depict classes and blue arrows depict object properties. Reused classes from well-known ontologies are depicted in white, e.g., `schema:Person`.

Reusing Well-known Ontologies

Various concepts and properties are reused from well-known ontologies. For adding specific annotation elements to SCORVoc, the vocabularies *skos* and *Dublin Core* are employed. The class `schema:Person` from the vocabulary *schema.org* is used to represent the person concept. This class is connected with the *scor:Skill* class to describe a certain person in the Supply Chain having certain skills.

Ontology Design Patterns

```
@prefix scor : <http://purl.org/eis/vocab/scor#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix olo: <http://smiy.sourceforge.net/olo/spec/orderedlistontology.html> .

scor:PlanReturnStepList  a  olo:OrderedList ;
    dcterms:title        "Default Plan Return Steps"@en ;
    dcterms:description  "An ordered list of all process steps involved in the Plan Return process."@an ;
    olo:length  4 ;
    olo:slot  [
        olo:index  1 ;
        olo:item   scor:AssessAggregateReturnRequirements ;
    ] ;
    olo:slot  [
        olo:index  2 ;
        olo:item   scor:IdentifyAssessAggregateReturnResources ;
    ] ;
    olo:slot  [
        olo:index  3 ;
        olo:item   scor:BalanceReturnResourcesWithReturnRequirements ;
    ] ;
```

```
    olo:slot  [
        olo:index  4 ;
        olo:item   scor:EstablishCommunicateReturnPlans ;
    ] .
```

Listing 5.12: **Default Plan Return Steps.** The four ordered steps which belong to the Plan Return process are modeled by means of the Ordered List Ontology.

We analyze existing ODPs that can be applied to the modeling of SCORVoc. A pattern to be included is the transitive version of the `partOf` ODP. The objective is to connect SCOR processes in a transitive way as part of the Supply Chain. This option for part-whole relation allows for recursively getting all participant processes as subprocesses of major SCOR processes. The *sequence ODP*[25] is a pattern of interest in the SCOR domain. SCOR level three processes, also called *steps*, have an order of execution of these processes. Therefore, we following a more fine-grained implementation of the Sequence ODP which is available in the class `olo:OrderedList` from the Ordered List Ontology.[26] The objective with this pattern is to express the ordered list relation in SCORVoc (cf. Listing 5.12). Likewise, 30 instances of the class `olo:OrderedList` are created to represent the processes with similar characteristics.

Formalizing SCOR Metrics

Metrics are defined by SCOR to evaluate Supply Chains on certain aspects such as reliability or responsiveness. Similar to the aforementioned processes, the SCOR reference organizes metrics into a hierarchical structure, level 1 being the highest, and level 3 the lowest. SCOR provides for each metrics in a level a calculation plan that takes as an input the value of certain metrics at a lower level. In the following, one selected metric for each category is described[27].

Reliability Level 2 metric *Orders Delivered in Full* (RL 2.1) of performance indicator *Reliability* measures whether orders are received by the customer in the quantities committed. Its calculation plan is:

$$\frac{\#Orders\ delivered\ in\ full}{\#Orders\ delivered} * 100\%$$

An order is considered as delivered in full once it contains the correct items (RL 3.33, level 3 metric) with the correct quantity (RL 3.35, level 3 metric).

Responsiveness Level 1 metric *Order Fulfillment Cycle Time* (RS 1.1) measures the average cycle time in days it requires to achieve customer orders. Its calculation plan is:

$$\frac{\sum Actual\ Cycle\ Times\ for\ All\ Orders\ Delivered}{\#Orders\ Delivered}$$

It depends on other metrics of level 2 such as the time it takes to procure goods and services (RS 2.1), its production time (RS 2.2), the delivery, and the delivery retail time (RS 2.3).

[25] http://ontologydesignpatterns.org/wiki/Submissions:Sequence
[26] http://smiy.sourceforge.net/olo/spec/orderedlistontology.html
[27] For reference, we added SCOR identifiers in parentheses where applicable.

Agility The metric *Upside Supply Chain Flexibility* (AG 1.1) counts the number of days to achieve an unplanned increase (20% suggested by the reference) in the output of Source, Make and Deliver components.

$$\max(Source, Make, Deliver, SReturn, DReturn)$$

By assuming the production can run concurrently (one strategy provided by SCOR), it requires to identify the process (see metrics AG 2.1-5) within the enterprise whose adaption takes the most time.

Costs The metric *Production Cost* (CO 2.004) accounts for all costs involved in the production process.

$$\sum Labor + Automation + Property + Inventory$$

Thus, it depends on metrics which assemble the labor costs (CO 3.014), the automation costs (CO 3.015), the property, plant and equipment costs (CO 3.016) and the governance, risk, compliance, inventory and overhead costs (CO 3.0017).

Assets The metric *Cast-to-Cash Cycle Time* (AM 1.1) represents the time it takes for an enterprise to earn money on raw material investments.

$$\sum SalesOutstanding + Inventory - PayableOutstanding$$

Thus, it is necessary to summarize the days between a sale is made and the cash is received (AM 2.1) with the days of sales they are in the inventory (AM 2.2). That sum needs to be subtracted with the days between purchasing raw materials and their actual payment (AM 2.3).

Previous approaches defined a concept for each metric. However in SCORVoc, metrics are represented as data properties, and their calculation plan is represented as a SPARQL query. SCOR metrics are hence 'operationalized', in the sense that all information required for computing the metric is made available in an interoperable way. The metrics are then translated into queries operating on this information, i.e., SPARQL queries [**harris2013sparql**] and returning the respective KPI. The level 3 metrics are the *data capture* entry point in SCORVoc. They are hence defined as data properties. Their `rdfs:domain` points to their respective processes, given by SCOR. Their range is `xsd:decimal` since they all describe a number between 0-100 in percent values. Listing 5.13 shows an example for the full definition of the property `hasMetricRL_33`. Equally as for the documentation of processes, we expressed each property with the definition and the additional information provided by SCOR.

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix xsd:  <http://www.w3.org/2001/XMLSchema#> .
@prefix scor: <http://purl.org/eis/vocab/scor#> .

scor:hasMetricRL_33  a  owl:DatatypeProperty ;
    rdfs:label       "Delivery Item Accuracy"@en,
                     "Exactitud en Entrega de Items"@es;
    rdfs:comment     "Percentage of orders ..."@en ;
    skos:notation    "RL.3.33" ;
    rdfs:domain      scor:ItemAccuracyProcesses ;
    rdfs:range       xsd:decimal .
```

Listing 5.13: **Definition of a SCOR metric**

Table 5.5: **Summary of SCORVoc Characteristics.** The table shows a summary of the SCOR vocabulary in aspects such as general details, reused ontologies, documentation, naming conventions, multilinguality, availability, and used methodology.

General	**Name:**	SCOR Vocabulary (scor)
	Size	285 classes, 5 object properties, 249 data properties, 224 individuals
	DL Expressivity	ALCHO (D)
Reuse	**P-R1, Reused Ontologies**	dcterms, schema.org, Ordered List Ontology
Documentation	**P-Do1, P-Do3**	By means of `skos:definition`, `skos:altLabel`, `skos:prefLabel`
Naming Conventions	**P-N1 - P-N8**	CamelCase notation
Multilinguality	**P-M1, PM2**	English for all terms and german labels for core elements
Availability	**PersistentURI**	`http://purl.org/eis/vocab/scor#`
	GitHub	`https://github.com/vocol/scor`
	Licence	Creative Commons
	VoCol Instance	`https://vocol.iais.fraunhofer.de/scorvoc/`
Methodology	**Ont. Eng. Methodology**	King and Uschold [176]

Then, the level one and two metrics are formalized as SPARQL queries. When triggered, they compute the value of the metric using the values of lower level metrics.

5.4.6 Summary of Ontology Characteristics

Table 5.5 reports on the summary of characteristics for SCORVoc. As illustrated in Section 5.4.5 and Section 5.4.5, each concept contains a definition together with a full documentation based on the descriptions provided by SCOR.

5.4.7 Use Cases and Evaluation

With objective of evaluating the applicability of SCORVoc we first define SCOR metrics as SPARQL queries to simulate a business scenario. The knowledge of these metrics is considered as a major competitive advantage in the enterprise world. Besides the previous data example, we present and briefly discuss how the metrics are realized as SPARQL queries. For this purpose, we developed a SCOR test data generator and measured the execution time of typical queries.

```
@prefix scor: <http://purl.org/eis/vocab/scor#> .


scor:process_1  scor:isSubjectOf  "Keyboard X" ;
    scor:hasTimeStamp    "2015-10-01T12:52:16" ;
    scor:hasSupplier     scor:Logitech  ;
    scor:hasCustomer     scor:Dell .

scor:process_1 a scor:SourceStockedProduct ;
    scor:hasMetricRL_33  100  ;
    scor:hasMetricRL_50  90 .
```

Listing 5.14: **Example of data expressed using SCORVoc**

Listing 5.14 demonstrates an example of data expressed using SCORVoc. The example describes a scenario, i.e., `scor:process_1` in which goods, e.g., keyboards are received by an enterprise on a certain date. These goods are forwarded to the warehouse which is the reason for the classification of the process as a `scor:SourceStockedProduct`. Alternatively, if these goods are directly used in the production or for specialized client orders, the process can be classified as `scor:SourceMakeToOrderProduct` or `scor:SourceEngineerToOrderProduct`. This enables factories to distinguish more easily between possible unnecessary orders, which end up as *dead capital* in the stock. As a next step, all information related to this event is captured. The metric `scor:hasMetricRL_33` represents the accuracy of items and `scor:hasMetricRL_50` the quantitative accuracy. Thus, the example shows that this order achieved an accuracy of 90%.

```
PREFIX scor: <http://purl.org/eis/vocab/scor#> .
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#> .

SELECT  ( (xsd:decimal(?full)  / ( xsd:decimal (?notFull) )  *  100)  as  ?result)
WHERE   {  { SELECT ( (count(?deliveredInFull) )  as  ?full)
WHERE   {  ?deliveredInFull  scor:hasMetricRL_33  100 .
           ?deliveredInFull  scor:hasMetricRL_50  100 . }}}
SELECT  (  (count(?allDeliveries) )  as  ?notFull)
WHERE   {  ?allDeliveries  a  scor:Process . }}}
```

Listing 5.15: **Orders Delivery in Full metric**

Once the Supply Chain related information is captured using SCORVoc, the execution of SPARQL query metrics becomes feasible. Listing 5.15 shows the SPARQL query for calculating the *Perfect Order Fulfillment* metric. The query compares all complete deliveries achieving 100% with all deliveries in total by relying on the respective properties. Applied on the previous example, it returns 0% due to the delivery being incomplete. Listing 5.15 presents the *Orders Delivered In Full* metric. Orders are considered to be delivered entirely by SCOR once their item accuracy (RL 33) and quantitative accuracy (RL 50) correspond to 100%. Thus, every order below that value is regarded as incomplete. The *Order Fulfillment Cycle Time* is calculated by collecting the respective sum (days) of all sources (RS 21), make (RS 22), deliver (RS 23), and deliver retail (RS 24) processes and divides them by the amount of all orders.

```
PREFIX scor: <http://purl.org/eis/vocab/scor#> .
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#> .

SELECT  ( (xsd:decimal (?actualTime)) /
          (xsd:decimal (?allOrders))  as  ?result)
WHERE   { {SELECT  ( SUM(xsd:decimal(?value) )  as  ?actualTime)
WHERE   { ?order  scor:hasMetricRS_21
          |scor:hasMetricRS_22
          |scor:hasMetricRS_23
          |scor:hasMetricRS_24  ?value . }}{
SELECT  ( count (?order)  as  ?allOrders)
WHERE   { ?order a scor:Process . }}}
```

Listing 5.16: **Order Fulfillment Cycle Time metric**

For the calculation of the *Upside Supply Chain Flexibility* metric, it is necessary to gather the value of all flexibility properties, i.e., AG1-5 and identify the maximum value among them. Similar to a team is only as strong as its weakest part, a Supply Chain is only as agile as its slowest link.

```
PREFIX scor: <http://purl.org/eis/vocab/scor#> .
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#> .

SELECT  (  MAX (xsd:decimal (?flexibility) )  AS  ?result)
WHERE   {  ?order  scor:hasMetricAG_1
           |scor:hasMetricAG_2
           |scor:hasMetricAG_3
           |scor:hasMetricAG_4
           |scor:hasMetricAG_5  ?flexibility . }
```

Listing 5.17: **Upside Supply Chain Flexibility metric**

The *Production Cost* metric (CO 2.004) is dependent on the sum of the metric properties for the Costs in Labor (CO 14), Automation (CO 15), Property (CO 16), and Inventory (CO 17).

```
PREFIX scor: <http://purl.org/eis/vocab/scor#> .
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#> .

SELECT  (  SUM (xsd:decimal (?costs) )  AS  ?result)
WHERE   {  ?order  scor:hasMetricCO_14
           |scor:hasMetricCO_15
           |scor:hasMetricCO_16
           |scor:hasMetricCO_17  ?costs . }
```

Listing 5.18: **Production Cost metric**

Listing 5.19 presents the *Cast-to-Cash Cycle Time* metric (AM 1.1). The query selects the average time raw materials stays in inventory (AM 2) together with the time the payment is due to by the company (AM 1) subtracted by that of the customers (AM 3).

```
PREFIX scor: <http://purl.org/eis/vocab/scor#> .
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#> .

SELECT  (  AVG  (xsd:decimal  (?inventoryDays))
        +  AVG  (xsd:decimal  (?salesOutstanding))
        -  AVG  (xsd:decimal  (?payableOutstanding))  as  ?result)
WHERE   {  ?order  scor:hasMetricAM_1  ?salesOutstanding .
           ?order  scor:hasMetricAM_2  ?inventoryDays .
           ?order  scor:hasMetricAM_3  ?payableOutstanding . }
```

Listing 5.19: **Cash-to-Cash Cycle Time metric**

We demonstrate the feasibility of the approach on a SCOR dataset. To the best of our knowledge, the only available dataset for Supply Chains based on SCOR is *SCORmark*[28], which is assembled by a major consulting firm, but is only available for business customers and not open for research. We hence developed a open-source synthetic test data generator, available on GitHub.[29] The generator allows performing a round-trip between data representation and KPI evaluation. The testbeds permit assessing the performance of a SCORVoc implementation in a systematic and repeatable way. The generator creates data based on a number of parameters, i.e., Supply Chain depth, industry and Supply Chain partners. The Supply Chain depth sets the level from one main OEM enterprise to it suppliers' supplier. The industry generates plausible product lines and enterprise names. The Supply Chain partners determine the width of the Supply Chain. A minimum of two generates a binary tree to both sides. Various dataset sizes can be generated in order to assess the scalability as well as the correctness of the metric SPARQL query implementations. We evaluated the metrics for datasets which contain 100k, 500k, 1M and 2M triples. Table 5.6 presents an overview of the generated data for the 100k scenario. While the instances represent different types of processes, the related properties are mostly 3-level data type properties which are required by the metrics, e.g., `scor:hasMetricRL_50`. The values randomized within a certain range, e.g., >80% for Reliability. The queries are executed using the *ARQ* SPARQL processor. The machine we used for the experiment contains 8GB of RAM, 256GB SSD and an Intel i7-3537U CPU with 2.00GHz.

Overall this evaluation shows that the approach of having an executable vocabulary is feasible. Furthermore, the SCOR data generator can be used to systematically assess and evaluate SCORVoc compliant software solutions. Such solutions could be specific supply network

[28] http://www.apics.org/sites/apics-supply-chain-council/benchmarking
[29] https://github.com/vocol/scor/generator

Processes	Instances	Related Properties
Source	1.481	28.576
Make	1.785	23.216
Deliver	2.083	22.917
Plan	1.538	18.462

Table 5.6: **Generated data for the 100k scenario.** The table shows the values of the generated synthetic data; instances and properties values are listed w.r.t. to the main processes described by SCOR.

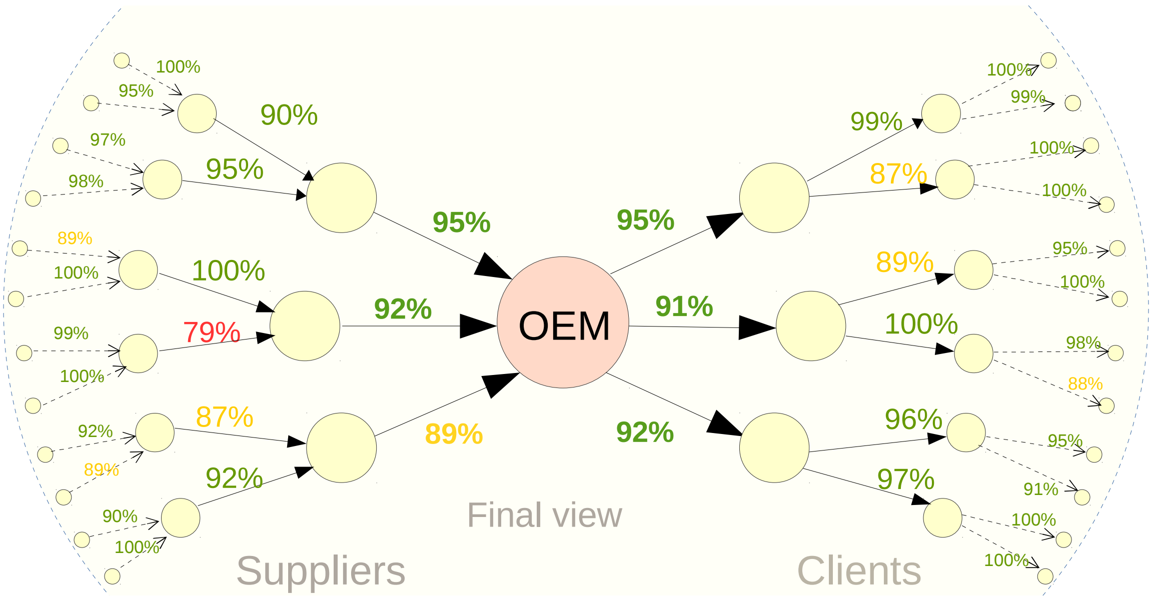


Figure 5.12: **Final view on the Supply Chain, where KPI information is propagated through the network.** With a semantic description of the communication in the Supply Chain, a factory possess a complete view of their customers and suppliers. Therefore, this fact empower the horizontal integration needed in I40 scenarios.

visualizations tools, Supply Chain robustness assessment frameworks, or scenario planning tools. In contrast with Figure 5.10, Figure 5.12 represents the full view on the entire Supply Chain.

5.5 Concluding Remarks

In this chapter, we investigated the development and application of ontologies for Industry 4.0 scenarios. To this end, a practical methodology for building ontologies of standards in Industry 4.0 scenarios is presented. This methodology is based on the Ushold and Kind as well as in the VoCol methodology and environment and present concrete practices to be applied in this kind of settings. Additionally, relevant ontologies to be reused are proposed such as the Ontology of Units of Measurements or Semantic Sensor Network ontology among others. Furthermore, three main standards covering representative areas in Industry 4.0 are researched, i.e., the RAMI4.0 architecture model with the Administration Shell concept, the AutomationML standard, as well as the Supply Chain Operation Reference. We have applied the proposed methodology to build ontologies for each one of these standards.

Firstly, the *RAMI4.0* ontology and the associated Administration Shell concept have been semantically described. Further, the hierarchy levels of the RAMI 4.0 model are modeled in an ontology. This permits to provide common descriptions of I40 components along with different types of data represented by various standards applied in the domain. One of the main advantages of the RAMI4.0 ontology is the uniform data representation according to the RDF model, which allows integrating and querying data described by in the Administration Shell. We demonstrated the applicability of the RDF-based approach by implementing it in a real-world use case, where we aligned the RAMI4.0 ontology with sensor data from a legacy system.

Secondly, we developed the AutomationML ontology that covers the AutomationML data exchange standard in the industrial engineering domain. AMLO provides concepts to support the design of an engineering system. This is the case of components and subcomponents, required and supported functionality of the components, various attributes, e.g., mechanical, electrical of logical ones, logical and physical connections between the system elements. The ontology design process is based on domain-specific and ontological requirements that are identified for the context of the AML standard. Particular attention during the design of AMLO is given to best practices for ontology design. The resulting ontology covers completely the XSD schema for AML and provides means for enhancing the engineering data with additional resources, e.g., by including the Ontology of Units of Measure. We also showed how AMLO can be used in real-world scenarios to improve the engineering processes during system design.

Finally, the SCORVoc vocabulary is designed, that formalizes and operationalizes the SCOR standard. We used an innovative methodology to develop the SCORVoc vocabulary, and provided means to automatically compute typical KPIs. Further, comprehensive test scenarios along with synthetic test cases for SCORVoc are described and implemented. SCORVoc, together with the formalized SPARQL queries, represents a comprehensive approach to facilitate information flows in Supply Chains, and enables the design of SCOR compliant IT applications. SCORVoc may help identify such limitations, and accompany the improvement of the SCOR specification.

To conclude, we demonstrate the benefits of the semantic representation of I40 entities by means of ontologies. Common use cases of the semantic representation in I40 scenarios are developed, e.g., the units of measurements. The representation and discovery of semantic heterogeneity conflicts among entities in these scenarios are introduced. Furthermore, the resolution of conflicts by considering and applying the semantics of the ontologies is developed. The knowledge graph approach for representing and linking entities poses many advantages for the realization of the I40 vision. The flexible schema representation, the creation of global unique identifiers for entities, the ease creating a multilingual representation, are some of these advantages that we can observe in the proposed approach.

CHAPTER 6

Integrating Cyber-Physical Systems into Knowledge Graphs

The engineering process in smart manufacturing environments combines various disciplines for designing and developing a CPS, i.e., mechanical, electrical, and software engineering. Despite using the same standard, e.g., AML, these disciplines have different views on top of the same CPS design. These perspectives comprise semantic interoperability conflicts among the disciplines that need to be conciliated [217]. The goal of the engineering process in smart manufacturing is to produce a final design where overlapping and inconsistencies are minimized and semantic heterogeneity conflicts are solved [122, 218, 219]. The final design has to respect the original intent of the different perspectives; it also has to ensure that all the knowledge encoded in each perspective is captured during the integration process. Typically, standards used for capturing the knowledge of this engineering process are XML-based standards. XML-based standards, e.g., AML, lack a formal semantic basis which is increasingly necessary to resolve semantic interoperability conflicts in industrial projects [26, 27]. In order to meet this need, the combination of knowledge graphs and rule-based approaches, capable to describe I40 scenarios while resolving semantic heterogeneity conflicts are needed.

Problem statement. In this chapter, we examine the problem of integrating CPS perspectives while resolving semantic heterogeneity conflicts; the characteristic uncertainty in CPS design environments is considered. This chapter addresses the third level in the general proposed contributions presented in this thesis, i.e., describing and integrating CPS into knowledge graphs (cf. Figure 6.1). Knowledge graphs are created and exploited by rule-based approaches to identify and resolve semantic interoperability conflicts between CPS perspectives.

The following research question is covered in this chapter:

> **RQ3**: How can existing rule-based approaches be utilized to resolve semantic interoperability conflicts over knowledge graphs?

Proposed solution. In this chapter, we tackle the problem of integrating CPS into knowledge graphs by applying Deductive Databases and Probabilistic Soft Logic approaches. The specifications of these formalizations are implemented in Alligator and SemCPS, respectively. These approaches enable the integration of CPS perspectives into knowledge graphs. The first one, Alligator, uses Datalog rules to express the knowledge of the perspectives in CPS design and perform the identification and solution of semantic interoperability conflicts. The second one, SemCPS, exploits the probabilistic soft logic framework to represent and integrate the

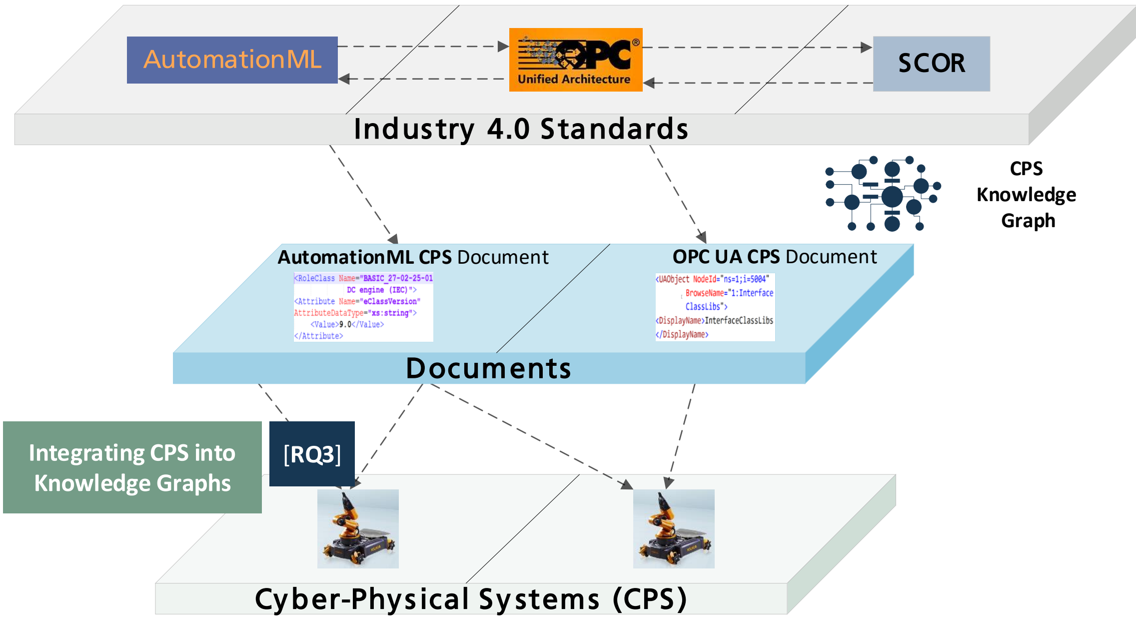


Figure 6.1: **The levels of the research problem addressed in this chapter.** We investigated challenges concerning the semantic integration of CPS into knowledge graphs.

knowledge of the CPS design perspectives into knowledge graphs. The current chapter is based on the following publications [211, 220–222].

In this chapter, the following crisp contributions that address **RQ3**, are presented.

- The formalization of the problem of identifying and resolving conflicts among I40 entities from different CPS perspectives following two logical approaches: the Deductive Databases and the Probabilistic Soft Logic. The specifications of these formalizations are implemented in ALLIGATOR and SEMCPS, respectively.
- An empirical study of the effectiveness of both approaches ALLIGATOR and SEMCPS, as well as its comparison with related approaches such as EDOAL and SILK.

6.1 Motivating Example

CPS are designed according to various engineering perspectives, e.g., specifications of a conveyor system according to mechanical, electrical and software viewpoints; a CPS final design includes the characteristics of the CPS specified in each perspective. However, perspectives are defined independently and conflicting specifications of the same entities may exist [5], e.g., a software perspective may specify components of a conveyor system that are not considered in the electrical viewpoint. These particularities the perspectives generate semantic heterogeneity conflicts, e.g., **SIC6**. Consequently, one of the biggest challenges for the realization of CPS is the integration of these perspectives based on the *knowledge* encoded in each of them [119, 120, 223]. A key challenge in such settings is the integration of multiple pieces of knowledge described in the same standard while resolving semantic interoperability conflicts. On one hand, perspectives enclose core characteristics of a CPS that need to be represented in the integrated design, e.g.,

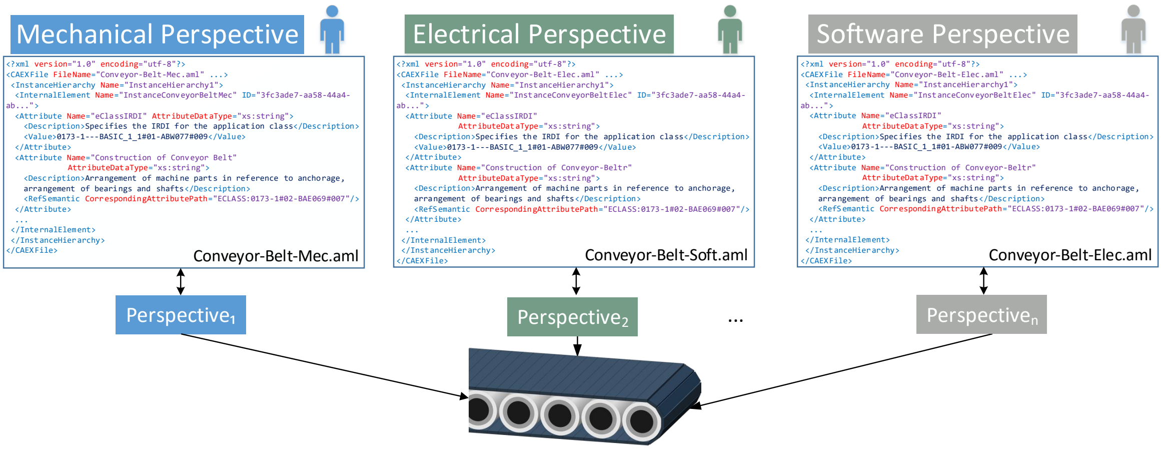
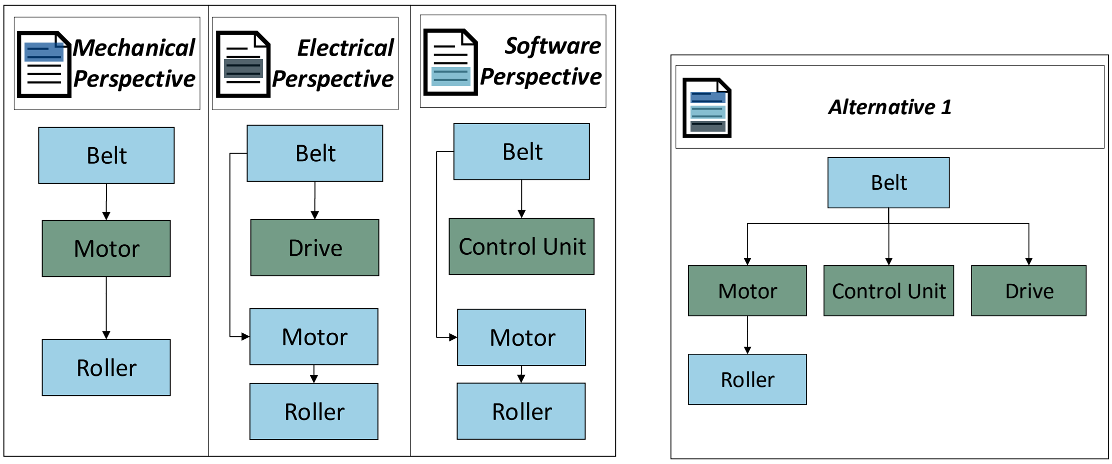


Figure 6.2: **Design of a Conveyor belt from different perspectives.** A Conveyor belt is considered as a CPS. The figure shows the result of an engineering process where a Conveyor belt is modeled from different perspectives: mechanical, electrical, and software perspectives. Equivalent entities of the Conveyor belt are defined differently in the perspectives, resulting in semantic interoperability conflicts.

descriptions of a Conveyor belt inputs and outputs and its main functionality; these characteristic correspond to *hard knowledge facts*. On the other hand, properties individually modeled in each perspective as well as the resolution of the corresponding heterogeneity conflicts that may be caused, need to be part of the final design according to how consistent they are with respect to the rest of the perspectives. These features are *uncertain* in the integrated CPS, e.g., entities expressed at the electrical perspective that are also included by the software perspective. These entities that are *totally* or *partially* covered by other perspectives can be modeled as *soft knowledge facts* in the integrated design. A number of approaches have been defined for integrating CPS perspectives [131, 223]. Although existing approaches support the integration of CPS perspectives based on the resolution of semantic heterogeneity conflicts, none of them is able to distinguish hard and soft knowledge facts during integration.

Figure 6.2 illustrates the results of an engineering process where a *Conveyor belt* is modeled from mechanical, software and electrical perspectives. Mechanical engineers design the components of a Conveyor belt from the mechanical perspective; electrical engineers model the electrical wiring topology inside the Conveyor belt whereas software engineers are focused on developing the system control for the Conveyor. AML is utilized in the perspectives to describe the entities that form the Conveyor belt. However, because physical structures in these perspectives are modeled with different properties, semantic interoperability conflicts might arise when integrating these designs.

Figure 6.3 delves into the CPS designed from the different perspectives. In this Figure, a scenario of CPS design without taking uncertainty into account is described. This means that in the three perspectives, there exist no soft knowledge facts to be inspected for obtaining the final integrated design. There are some entities and relations in each one of the perspectives that have to be respected when obtaining the final integrated design. For instance, in the mechanical perspective the relation between `Belt` and `Motor`; in the electrical perspective the relation between `Belt` and `Drive`; whereas in the software perspective the relation between `Belt` and `Control Unit`. Those entities are considered constraints in each one of the perspectives and



(a) Perspectives in CPS design (b) Alternative for the final CPS design

Figure 6.3: **Design of a Conveyor belt from different perspectives with hard knowledge facts.** The result from the engineering design of a Conveyor belt presented in Figure 6.2. (a) Different characteristics of the three perspectives; the same CPS is defined in terms of various entities and attributes with only hard knowledge facts. (b) An alternative for the final CPS design.

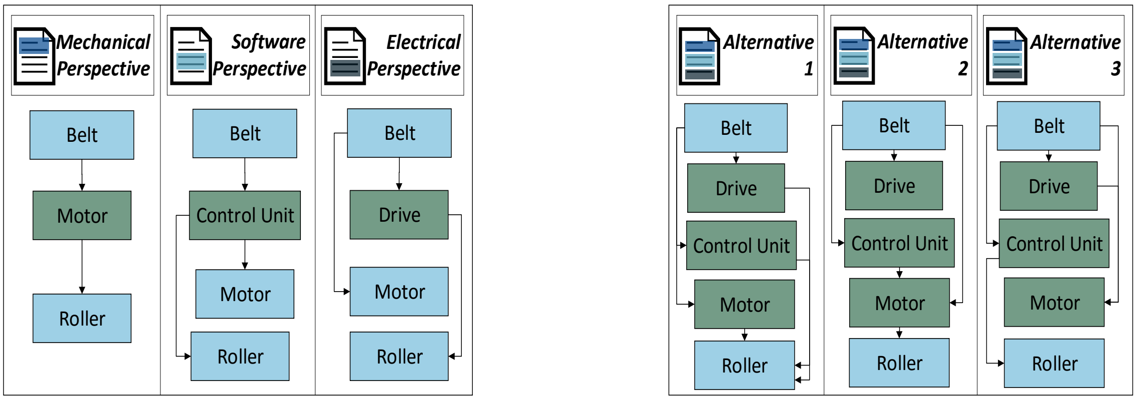
are required to be included in the final integrated design of a CPS; they correspond to hard knowledge facts. In addition to these relations, the three perspectives comprise the relation between the entities `Motor` and `Roller`. In this example, there exist no overlap between the relations of the perspectives. Therefore, it comprises only hard knowledge facts to be considered.

Figure 6.4 depicts three perspectives of the CPS in Figure 6.2; they share some entities, e.g., `Belt`, `Motor`, and `Roller`. In this example, the hard knowledge facts are the same than those in Figure 6.3. Moreover, some relations are only considered in one of the perspectives, e.g., the relation between `Control Unit` and `Roller` and between `Drive` and `Roller`. In addition, the same entities are modeled at different levels, e.g., `Motor`, which is placed in the second level in the mechanical perspective and in the third level in the software and electrical perspectives. These entities are uncertain in the final design and could be represented as soft knowledge facts. Accordingly, these differences in the modeling of the same CPS cause semantic interoperability conflicts of Representation (**SIC4**), Granularity (**SIC6**), and Missing Item (**SIC7**).

Figure 6.5 dives into the general example presented in Figure 6.4. In this case, a Conveyor belt is described by the mechanical and electrical perspectives in terms of the AML standard. Despite entities are identified by means of the eCl@ss properties, different views exist on top of the same Conveyor belt by the two disciplines, i.e., two perspectives of the same CPS. Thus, semantic interoperability conflicts of Schema (**SIC2 - M3**) occur between the two perspectives.

6.2 Problem Definition

In this section, we describe the problem of semantically integrating different perspectives of CPS design. In order to integrate the CPS perspectives, the subproblems of *CPS conflict identification* and *CPS conflict resolution* among entities of CPS are also investigated. To overcome the presented problem, we rely on the creation and exploitation of knowledge graphs.



(a) Perspectives in CPS design

(b) Alternatives between perspectives of a CPS

Figure 6.4: **The design of a Conveyor belt from different perspectives generates divergent alternatives**. The result from the engineering design of a Conveyor belt presented in Figure 6.2. (a) Different characteristics of descriptions of the three perspectives; the same CPS is defined in terms of various entities and relations. (b) Various proposed alternatives for the final CPS design.

```
<CAEXFile FileName="ConveyorBeltMec.aml" ...>
 <InstanceHierarchy Name="ConveyorBelt">
 <InternalElement ID="45e20f8e" Name="Belt">
  <Attribute Name="Motor">
    <RefSemantic AttributePath="ECLASS:Motor.."/>
     <Attribute Name="Roller">
     <RefSemantic AttributePath= "ECLASS:Roller.."/>
   </Attribute>
  </Attribute>
 </InternalElement>
 </InstanceHierarchy>
</CAEXFile>
```

(a) **Mechanical Perspective**

```
<CAEXFile FileName="ConveyorBeltElec.aml" ...>
 <InstanceHierarchy Name="ConveyorBelt">
 <InternalElement ID="45e20f8e" Name="Belt">
  <Attribute Name="Motor">
    <RefSemantic AttributePath="ECLASS:Motor.."/>
  </Attribute>
  <Attribute Name="Drive">
    <RefSemantic AttributePath= "ECLASS:Drive.."/>
     <Attribute Name="Roller">
     <RefSemantic AttributePath= "ECLASS:Roller.."/>
    </Attribute>
  </Attribute>
 </InternalElement>
 </InstanceHierarchy>
</CAEXFile>
```

(b) **Electrical Perspective**

Figure 6.5: **Example of the Mechanical and Electrical perspectives of a CPS described in the AML standard**. A Conveyor belt is described in terms of the AML standard by the Mechanical and Electrical perspectives. Despite entities are identified by means of the eCl@ss properties, different views exist on top of the same Conveyor belt by the two disciplines, i.e., two perspectives of the same CPS. Thus, semantic interoperability conflicts of Schema (**SIC2 - M3**) occur between the two perspectives.

Definition 1 (CPS knowledge graph) *Given sets I and V that correspond to URIs identifying entities in a CPS document and terms from a CPS standard ontology*[1]*, respectively; furthermore, let L be a set of literals. A CPS Knowledge Graph $\mathcal{G}$ is a 4-tuple $\langle I, V, L, G \rangle$, where G is a set of triples of the form $(s, p, o) \in I \times V \times (I \cup L)$. Given two CPS knowledge graphs $\mathcal{G}_1 = \langle I, V, L, G_1 \rangle$, $\mathcal{G}_2 = \langle I, V, L, G_2 \rangle$ the entailment for $\mathcal{G}_1 \models \mathcal{G}_2$ is defined as the standard RDF entailment G_1 and G_2, i.e., $G_1 \models G_2$.*

[1] In general, V can refer to different ontologies representing standards, but in this work, we focus on the AML ontology.

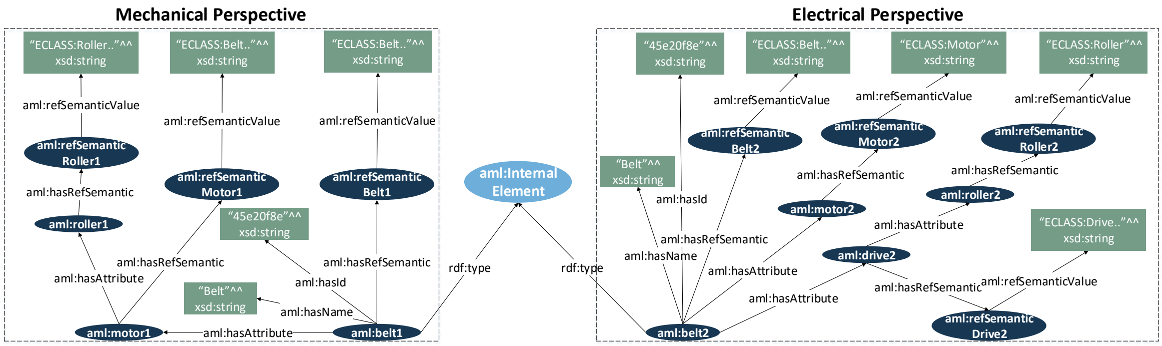


Figure 6.6: **RDF Graphs of the mechanical and electrical perspectives.** (a) RDF graph representation of the mechanical perspective depicted in Figure 6.5(a). RDF graph representation of the mechanical perspective depicted in Figure 6.5(b) (b). RDF graph representing CPS entities of the mechanical and electrical perspectives in Figure 6.5.

A CPS Knowledge Graph $\mathcal{G}_i = \langle I, V, L, G \rangle$ can represent information from one or several CPS documents D_i, where I is the set of URIs that identify the CPS entities in D_i, and the RDF graph G describes the relationships among the CPS entities in D_i.

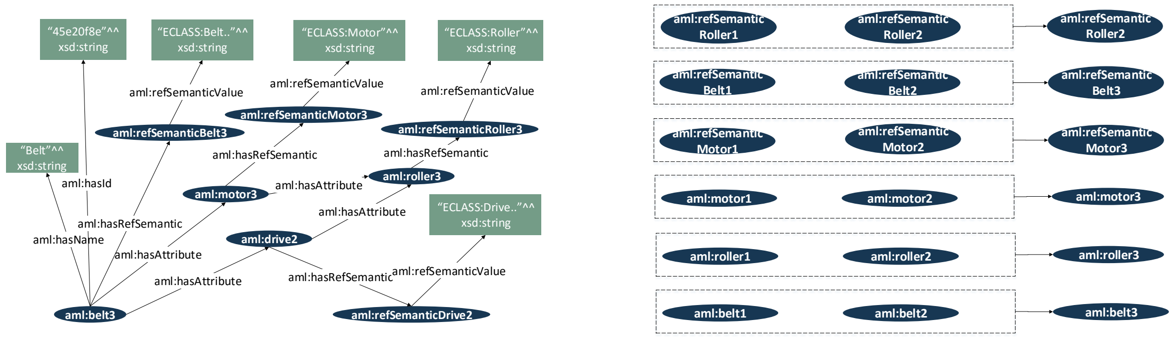
Example 1 *Consider the RDF graph G_1 in Figure 6.6. This graph comprises RDF resources representing the CPS entities in the mechanical and electrical perspectives shown in Figure 6.5; the CPS ontology is used to describe these resources. A CPS knowledge graph document $\mathcal{G}_1 = \langle I_1, V, L, G_1 \rangle$ formally describes this RDF representation of these two perspectives, where I_1 is the set of the resources in G_1, V is the CPS ontology, i.e., the AML ontology in this case, and L_1 the set of literals in G_1.*

Definition 2 (Ideal CPS knowledge graph) *Given a CPS knowledge graph $\mathcal{G} = \langle I, V, L, G \rangle$, there is an ideal CPS knowledge graph $\mathcal{G}^* = \langle I^*, V, L, G^* \rangle$ such that $\mathcal{G}^*$ comprises only conflict-free CPS entities. Additionally, there is a homomorphism $\sigma : \theta \rightarrow \theta^*$. The RDF ideal graph G^* is defined as follows:*

$$G^* = \{(\sigma(s),\ p,\ \sigma(o)) \mid (s,\ p,\ o) \in G\}$$

Example 2 *Consider the RDF graph in Figure 6.6. The CPS knowledge graph $\mathcal{G}^* = \langle I^*, V, L, G^* \rangle$ describes this RDF graph, where I^* is the set of RDF resources in the graph, V is the CPS ontology, L is the set of RDF literals in the graph, and G^* is this RDF graph. $\mathcal{G}^*$ represents the ideal conflict-free CPS knowledge graph of $\mathcal{G}_1$. Figure 6.7(b) shows a homomorphism σ that maps two equivalent entities in the RDF graph in Figure 6.6 to the same entity in Figure 6.7(a).*

Definition 3 *Consider a CPS knowledge graph $\mathcal{G} = \langle I, V, L, G \rangle$, an ideal conflict-free CPS knowledge graph $\mathcal{G}^* = \langle I^*, V, L, G^* \rangle$, and a homomorphism $\sigma : I \rightarrow I^*$. A set of semantic interoperability conflicts in $\mathcal{G}$ with respect to $\mathcal{G}^*$ and σ, conflicts($\mathcal{G} \mid \mathcal{G}^*, \sigma$), corresponds to the set of CPS entity pairs (E_i, E_j) in $I \times I$ such that E_i and E_j are different but that σ maps to the same target CPS entity in I^*: conflicts($\mathcal{G} \mid \mathcal{G}^*, \sigma$)$=\{(E_i, E_j) \mid E_i, E_j \in I$ and $E_i \neq E_j$ and $\sigma(E_i) = \sigma(E_j)\}$.*



(a) Ideal RDF graph

(b) Homomorphism σ

Figure 6.7: **Ideal conflict-free CPS knowledge graph**. (a) An RDF graph where there is only one RDF resource for entities presenting semantic interoperability conflicts in the mechanical and electrical perspectives in Figure 6.2. (b) A homomorphism σ maps entities presenting semantic interoperability conflicts into the RDF graph in Figure 6.6 to the same resource in the ideal RDF graph.

Example 3 *Given CPS knowledge graphs $\mathcal{G}_1$ and $\mathcal{G}^*$ from Example 1 and Example 2, and the homomorphism σ in Figure 6.7(b). The set of conflicts($\mathcal{G}_1 \mid \mathcal{G}^*, \sigma$) corresponds to the set of pairs of RDF resources in the RDF graph in Figure 6.6 that σ maps to the same resource in the ideal RDF graph (cf. Figure 6.7(a)).*

Given a CPS knowledge graph $\mathcal{G} = \langle I, V, L, G \rangle$, the *CPS Conflicts Identification* problem determines if a pair (E_k, E_l) of CPS entities in I comprise semantic interoperability conflicts.

Definition 4 *Consider a CPS knowledge graph $\mathcal{G} = \langle I, V, L, G \rangle$, an ideal conflict-free CPS knowledge graph $\mathcal{G}^* = \langle I^*, V, L, G^* \rangle$, and a homomorphism $\sigma : I \rightarrow I^*$. The* CPS Conflicts Identification *problem corresponds to the problem of deciding if $(E_k, E_l) \in I \times I$ belongs to conflicts($\mathcal{G} \mid \mathcal{G}^*, \sigma$).*

Definition 5 *Consider a CPS knowledge graph $\mathcal{G} = \langle I, V, L, G \rangle$ and a set SC($\mathcal{G}$) of pairs of CPS entities comprising semantic interoperability conflicts in G. The problem of* CPS Conflict Resolution *corresponds to the problem of creating a CPS knowledge graph $\mathcal{G}' = \langle I', V, L, G' \rangle$ and a homomorphism $\sigma' : I \rightarrow I'$, such that:*

- *For each (E_i, E_j) in SC($\mathcal{G}$), there is a CPS entity E_m in I' such that $\sigma'(E_i) = \sigma'(E_j) = E_m$.*
- $G' = \{(\sigma'(s),\ p,\ \sigma'(o)) \mid (s,\ p,\ o) \in G\}$.

$\mathcal{G}'$ represents the CPS knowledge graph where pairs of CPS entities in SC($\mathcal{G}$) are represented as one RDF CPS entity.

6.3 A Deductive Database Approach

In this section, we outline an approach based on Deductive Databases. The presented approach aims at solving the problems of *CPS Conflicts Identification* and *CPS Conflicts Resolution*. Further, we present Alligator, which implements the proposed approach by relying on

Deductive Databases. ALLIGATOR employs Deductive Databases and the AML ontology for representing CPS documents, as well as for detecting semantic heterogeneity conflicts whenever CPS perspectives need to be integrated.

The solution of the *CPS Conflicts Identification* and *CPS Conflicts Resolution* problems require the existence of the ideal conflict-free CPS document $\mathcal{G}^*$ and the homomorphism σ. However, in practice, neither $\mathcal{G}^*$ and σ are known, and ALLIGATOR computes an approximation of the problem. We use $SC(\mathcal{G})$ to refer to the set of pairs (E_k, E_l) that correspond to the solutions of this problem. Once a set $SC(\mathcal{G})$ of CPS entities presenting semantic interoperability conflicts in G is identified as the solution of the CPS *Conflicts Identification problem*, the problem of *CPS Conflicts Resolution* corresponds to the problem of creating a CPS knowledge graph where semantic interoperability conflicts in $SC(\mathcal{G})$ are solved.

The documents that correspond to the different perspectives of the same CPS are the input for the creation of the knowledge graph $\mathcal{G}$. $\mathcal{G}$ formally represents the union of these documents. Additionally, a set of Datalog extensional facts (EDB) representing the entities in the RDF document G is created. With the purpose of exploiting the knowledge encoded in such a knowledge graph, a set of Datalog intentional rules (IDB) is developed. These rules describe existing semantic interoperability conflicts that can occur among CPS documents. The goal of these rules is to compute the set $SC(\mathcal{G})$ from the Datalog representation of $\mathcal{G}$. $SC(\mathcal{G})$ is computed as the least minimal fixpoint of the Datalog rules in IDB and the facts in EDB [224].

Alligator Rule-based representation of Interoperability Conflicts One of the key innovations of ALLIGATOR revolves on the use of a Datalog-rule approach to effectively resolve semantic heterogeneity conflicts between CPS documents. For this purpose, we have developed a set of rules covering the main characteristics of a standard commonly used for describing CPS, i.e., AML. It is important to remark that the rules have been defined taking into account the object and datatype properties of the AML ontology.

With respect to AML attributes, there are cases in which they share the same id and name but are not semantically equivalent. On the contrary, even if two attributes are defined with different names, e.g., *Length* and *StrictLength*, they can still be semantically equivalent whenever they refer to the same eCl@ss identification value, i.e., eCl@ss IRDI. In detail, the AML entity `refSemantic` refers to the eCl@ss IRDI by means of `CorrespondingAttributePath` (cf. Figure 6.5 line 5).[2] In order to identify and resolve this semantic heterogeneity conflict, the rules presented in Listing 6.1 and in Listing 6.2 are defined.

```
sameRefSemantic(T,Z) ∧ hasRefSemantic(X,T) ∧ hasRefSemantic(Y,Z) ⇒ sameAttribute(X,Y)
```

Listing 6.1: **Semantic equivalence of two AML attributes**

```
hasCorrespondingAttributePath(X,Z) ∧ hasCorrespondingAttributePath(Y,Z) ⇒ sameRefSemantic(X,Y)
```

Listing 6.2: **Semantic equivalence of two semantic references**

To determine that two entities of type Role Classes are semantically equivalent according to their reference to eCl@ss, they have to contain the same version, classification and IRDI. To represent these three conditions, the rule described in Listing 6.3 is built.

[2] The AML CorrespondingAttributePath is shortened to AttributePath in Figure 6.5 for space reasons.

```
type(X,RoleClass) ∧ type(Y,RoleClass) ∧ sameEClassIRDI(A,B) ∧ sameEClassClassification(C,D) ∧
sameEClassVersion(E,F) ∧ hasAttribute(X,A) ∧ hasAttribute(X,C) ∧ hasAttribute(X,E) ∧
hasAttribute(Y,B) ∧ hasAttribute(Y,D) ∧ hasAttribute(Y,F) ⇒ sameRoleClass(X,Y)
```

Listing 6.3: **Semantic equivalence of two Role Classes**

The rule for identifying the semantic equivalence of two Role Classes (cf. Listing 6.3) relies on simpler rules such as the one defined in Listing 6.4. This rule, computes the equivalence of two `eClassIRDI` attributes. Similarly, we have defined rules to determine whether two values of `eClassVersion` and `eClassClassification` are semantically equivalent.

```
hasAttributeName(X, "eClassIRDI") ∧ hasAttributeName(Y,"eClassIRDI") ∧ hasAttributeValue(X,Z) ∧
hasAttributeValue(Y,Z) ⇒ sameEClassIRDI(X,Y)
```

Listing 6.4: **Semantic equivalence of two eClassIRDI attributes**

These rules correspond to a sample of the type of rules implemented in Alligator; the complete set of rules is available in Github.[3]

6.3.1 Alligator Architecture

The architectural components of Alligator are depicted in Figure 6.9. Given a set of CPS documents representing different perspectives, the *Alligator Data Model Creation* component generates a CPS knowledge graph $\mathcal{G} = \langle I, V, L, G \rangle$ that formally describes the union of these input CPS documents. Additionally, a set of EDB representing the triples in the RDF document G is created. The *Deductive System Engine* relies on the set of IDB to compute the set *SC($\mathcal{G}$)* from the Datalog representation of $\mathcal{G}$. *SC($\mathcal{G}$)* is computed as the least minimal fixpoint of the Datalog rules in IDB and the facts in EDB. Further, *SC($\mathcal{G}$)* is utilized by the *Integrated CPS Document Creation* component to resolve the *CPS Conflicts Resolution* problem, and to produce an integrated CPS document where CPS entities in *SC($\mathcal{G}$)* are integrated as one CPS entity.

Alligator Data Model and Deductive System Engine Alligator represents CPS documents as RDF graphs. CPS documents, described in the AML standard, are translated into RDF using Krextor [225], a XSLT-based framework for converting XML to RDF. The AML ontology is used to describe CPS entities and relations. Further, CPS documents are modeled as facts in an extensional database (EDB) of a Datalog program P; for each type of CPS entity in the standard exists an extensional Datalog predicate in P. Rules in the intensional database (IDB) of the Datalog program P characterize types of semantic heterogeneity conflicts. Intensional Datalog predicates represent semantic interoperability conflicts that can exist between the different CPS entities. The Alligator *Deductive System Engine* performs a bottom-up evaluation of P following a semi-naïve algorithm that stops when the least fixed-point is reached [224]. The intentional predicates inferred in the evaluation of P correspond to the pairs of semantic interoperability conflicts in the set *SC($\mathcal{G}$)*.

6.4 A Probabilistic Soft Logic Approach

In this section, we present the solution for the defined problem by relying on Probabilistic Soft Logic. For this purpose, CPS uncertainty knowledge graphs are defined. Then, the problem of integrating CPS perspectives is presented as an inference problem on uncertain knowledge

[3] https://github.com/i40-Tools/AlligatorRules

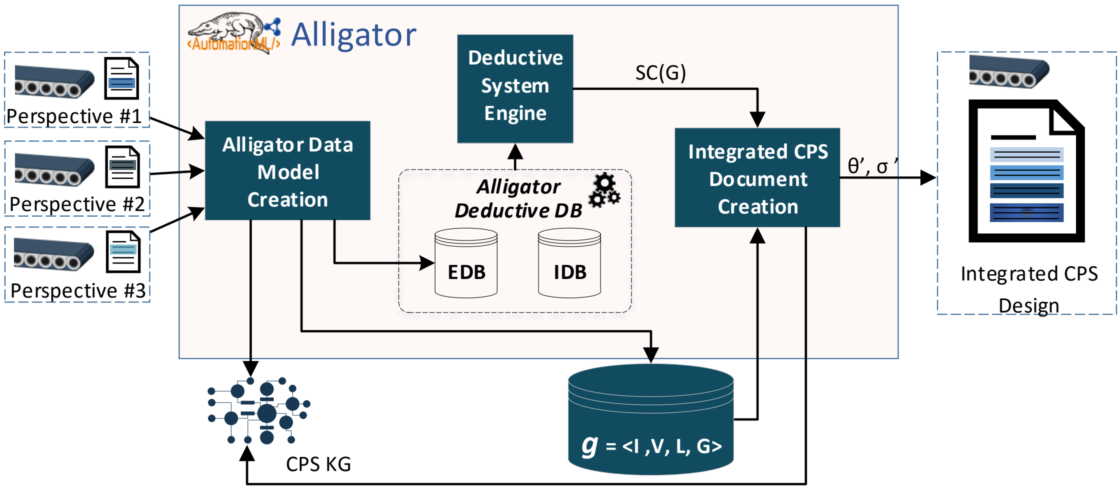


Figure 6.8: **The Alligator Architecture**. ALLIGATOR receives CPS documents describing different perspectives and creates an integrated CPS document. CPS documents are represented as RDF graphs and as Datalog predicates (EDB); Datalog intentional rules (IDB) characterize semantic heterogeneity conflicts. A bottom-up evaluation of the Datalog program identifies semantic interoperability conflicts between CPS documents describing different perspectives of CPS design.

graphs. The PSL framework provides a practical solution to this problem. The presented approach is implemented in SEMCPS. SEMCPS is a rule-base approach that relies on PSL for capturing the knowledge encoded in different CPS perspectives and exploiting this knowledge for CPS perspective integration. SEMCPS includes weighted rules representing the conditions to be met by hard and soft knowledge facts. SEMCPS rely on uncertain knowledge graphs [226, 227] where edges are annotated with weights to represent the knowledge of different perspectives and to integrate this knowledge into a final design. Chekol *et al.* [226] have shown that knowledge graphs can be extended with uncertainty; the maximum a-posteriori inference process from Markov Logic Networks (MLNs) is used to compute the interpretation of the triples in an uncertain KG that minimizes the overall uncertainty. Similarly, we define a *CPS Uncertain Knowledge Graph* as a knowledge graph where each fact is annotated with a weight in the range [0,1]; weights represent uncertainty about the membership of the corresponding facts to the knowledge graph, i.e., soft knowledge facts. A typical inference task in MLNs is to encounter a complete assignment to all ground atoms that maximize the probability, i.e., the most probable state of the world. In the problem we are tackling, the most probable state coincides with the most probable alternate design. Moreover, we devise an entailment relation between two CPS uncertain knowledge graphs; this relation allows for deciding when a CPS uncertain knowledge graph covers the hard and soft knowledge facts of the other knowledge graph.

Definition 6 (Uncertain Knowledge Graph) *Formally, given L, I, and V, three sets of literals, URIs identifying entities in a CPS document, and terms in a CPS standard ontology, respectively. A* CPS Uncertain Knowledge Graph $\mathcal{G}_u$ *is a 5-tuple* $\langle I, V, L, D, U \rangle$*:*

- *D is an RDF graph of the form $(s, p, o) \in I \times V \times (I \cup L)$. D represents a set of* hard knowledge facts.

- *U is an RDF graph where triples are annotated with weights. U is a set of* soft knowledge facts, *defined as follows:*

$$U = \{(t, w) \mid t \in I \times V \times (I \cup L) \text{ and } w \in [0, 1]\}$$

- *$\tau(U)$ is the set of triples in U, with $\tau(U) \cap D = \emptyset$, i.e.,*

$$\tau(U) = \{t \mid (t, w) \in U\}.$$

Example 4 *Figure 6.11(a) shows an Uncertain Knowledge Graph $\mathcal{G}_u1$ for Alternative1 in Figure 6.4(b). Edges between green nodes represent hard knowledge facts in D, while soft knowledge facts are modeled as edges between blue nodes in U. Entities in the perspectives in Figure 6.13(a) correspond to hard knowledge facts, e.g., entities stating that* `Belt` *is related to* `Motor`, `Control Unit`, *and* `Drive`. *Also, the relation between* `Motor` *and* `Roller` *is only included in one perspective; thus it corresponds to a soft knowledge fact in U.*

The semantics of a CPS uncertain KG $\mathcal{G}_u$ is defined in terms of the probability distribution of the values of weights of the triples in $\mathcal{G}_u$. As defined by Chekol *et al.* [226], the weights of the triples in $\mathcal{G}_u$ are characterized by a log-linear probability distribution. For any CPS Uncertain Knowledge Graph $\mathcal{G}_u^*$ over the same sets I, V, and L, i.e., $\mathcal{G}_u^*$=$\langle I, V, L, D^*, U^* \rangle$ the probability of $\mathcal{G}_u^*$ is as follows:

$$P(\mathcal{G}_u^*) = \begin{cases} \frac{1}{Z} exp \left(\sum_{\{(t_i, w_i) \in U : D^* \cup \tau(U^*) \models t_i\}} w_i \right) & \text{if } D^* \cup \tau(U^*) \models D \\ 0 & \text{otherwise} \end{cases} \tag{6.1}$$

Z is the normalization constant of the log-linear probability distribution P.

Example 5 *Consider the CPS uncertain KGs depicted in Figure 6.11; they represent alternate integrated designs in Figure 6.4(b). In Figure 6.12, we present a CPS uncertain KG $\mathcal{G}_u$ where all the entities in the three perspectives are included in the knowledge graph D, i.e., they correspond to hard knowledge facts; additionally, the knowledge graph U includes uncertain triples representing soft knowledge facts; weights denote how many times a fact is represented in the three perspectives. For example, the relation between* `Motor` *and* `Roller` *is only included in one out of three perspectives, so, the weight is 0.3. This KG can be seen as a complete integrated design of the CPS. Furthermore, uncertain KGs in Figure 6.11(a) and Figure 6.11(b) represent alternate integrated designs; the probability of these KGs with respect to the one in Figure 6.12 is computed following equation 6.1. Figure 6.11(a) presents a KG with the highest probability; it corresponds to* Alternative1 *in the motivating example where the majority of the facts in the KG are also included in the KG of Figure 6.12.*

Definition 7 *Let $\mathcal{G}_u$=$\langle I, V, L, D, U \rangle$ be a CPS uncertain knowledge graph. The entailment for any $\mathcal{G}_u^*$=$\langle I, V, L, D^*, U^* \rangle$ $\mathcal{G}_u^* \models_u \mathcal{G}_u$ holds if $P(\mathcal{G}_u^*) > 0$.*

Example 6 *Consider again the CPS uncertain KGs presented in Figure 6.11, because the probability of the uncertain KGs in Figure 6.11(a), and Figure 6.11(b) with respect to the KG in Figure 6.12 is greater than 0.0, we can say that the entailment relation is met, i.e., $\mathcal{G}_u^1 \models_u \mathcal{G}_u$, $\mathcal{G}_u^2 \models_u \mathcal{G}_u$, and $\mathcal{G}_u^3 \models_u \mathcal{G}_u$.*

Integrating CPS perspectives corresponds to the problem of identifying a CPS Uncertain KG $\mathcal{G}_u^*$ where the probability distribution with respect to the complete integrated design $\mathcal{G}_u$ is maximized. This problem optimization is follows:

$$\underset{\mathcal{G}_u^* \models_u \mathcal{G}_u}{\text{argmax}}(P(\mathcal{G}_u^*))$$

Example 7 *Consider the CPS uncertain KGs shown in Figure 6.12. The probabilities for the KGs representing the alternate designs with respect to* $\mathcal{G}_u$ *are as follows:* $\mathcal{G}_{u1} = 0.9$*,* $\mathcal{G}_{u2} = 0.6$ *and* $\mathcal{G}_{u3} = 0.6$*. An optimal solution of integrating CPS perspectives is the CPS uncertain KG in Figure 6.11(a); this KG represents* Alternative1 *which according to Prinz [218], is the most complete representation of the CPS perspectives described in Figure 6.13(a).*

SemCPS Rule-based representation of Interoperability Conflicts As shown by Chekol *et al.* [226], solving the maximum a-posteriori inference process required to compute the probability of an uncertain KG, is NP-hard in general. In order to provide a practical solution to this problem, we propose a rule-based system that relies on PSL to generate uncertain KGs that correspond to approximate solutions to the problem of integrating CPS perspectives.

The PSL program receives as input facts representing all the entities in the perspectives to be integrated. Next, rules are employed to determine equivalences between the entities, e.g., `Belt` is related `Motor` and `Motor` is related to `Roller`, then `Belt` is related to `Roller`.

```
Component(A) ∧ Component(B) ∧ hasRefSem(A,Z) ∧ hasRefSem(B,Z) ⇒ SemSimComp(A,B) | 0.8
```

Listing 6.5: **Semantic equivalence of two Components**

For example, Listing 6.5 shows a rule for generating new entities in an integrated design assuming that semantically similar components are related to same attributes. Based on the weights of rules, the facts have a high degree of membership in the integrated design. Specific knowledge about the standard, i.e., AML in this case, need to be encoded to identify semantic interoperability conflicts. This differentiation of importance for detecting the semantic between two entities is expressed in PSL by giving different weights to the rules. In Listing 6.6 we defined the rule that matches two entities based on its identification and name. A weight of 0.5 is set to this rule. This weight represents the importance of the combination of the identification and name for finding the semantic equivalence between two attributes.

```
hasAttributeID(A,Y) ∧ hasAttributeID(B,Z) hasAttributeName(A,N) ∧ hasAttributeName(B,M) similarValue(Y,Z) ∧
similarValue(N,M) ⇒ sameAttribute(A,B) | 0.5
```

Listing 6.6: **Semantic equivalence of two AML attributes by ID and Name**

In an ideal scenario, one can argue that with equivalent values for the name and identification of two attributes is sufficient to determine the equivalence between them.

```
hasRefSemantic(X,T) ∧ hasRefSemantic(Y,Z) ∧ sameRefSemantic(T,Z) ⇒ sameAttribute(X,Y) | 1.0
```

Listing 6.7: **Semantic equivalence of two AML attributes**

Nevertheless, the definition of the standard requires the same value for a semantic reference that points to the eCl@ss catalog. To encode this knowledge, Listing 6.7 presents a PSL rule for determining whether two attributes are equivalent. The rationale of this rule can be read as follows: if two attributes comprise the same values for the properties that link them to their semantic references, then the probability that these attributes are equivalent is high, i.e., 0.9. Similarly, to determine that two Role Classes are semantically equivalent according to their reference to eCl@ss they have to contain the same version, classification and IRDI. To represent these three conditions, the rule described in Listing 6.8 is built.

```
type(X,RoleClass) ∧ type(Y,RoleClass) ∧ sameEClassIRDI(A,B) ∧ sameEClassClassification(C,D) ∧
sameEClassVersion(E,F) ∧ hasAttribute(X,A) ∧ hasAttribute(X,C) ∧ hasAttribute(X,E) ∧
hasAttribute(Y,B) ∧ hasAttribute(Y,D) ∧ hasAttribute(Y,F) ⇒ sameRoleClass(X,Y) | 1.0
```

Listing 6.8: **Semantic equivalence of two Role Classes**

Listing 6.8 depicts the rule for identifying semantic equivalences of two Role Classes. This rule relies on simpler rules such as the one defined in Listing 6.9. This rule defines the equivalence of two `eClassIRDI` attributes. Similarly, we defined rules to determine whether two values of `eClassVersion` and `eClassClassification` are semantically equivalent.

```
hasAttributeName(X, "eClassIRDI") ∧ hasAttributeName(Y,"eClassIRDI") ∧ hasAttributeValue(X,Z) ∧
hasAttributeValue(Y,Z) ⇒ sameEClassIRDI(X,Y) | 0.9
```

Listing 6.9: **Semantic equivalence of two eClassIRDI attributes**

The PSL program builds the uncertain KG in Figure 6.11(a) maximizing the probability distribution with respect to the complete integrated design in Figure 6.12.

6.4.1 SemCPS Architecture

We present SemCPS, an approach to integrate different perspectives of a CPS. Figure 6.9 depicts the architectural components of SemCPS. SemCPS receives as input a set of documents describing a CPS in a given smart manufacturing standard and a *membership degree threshold*; the output is a final integrated design of the CPS. SemCPS builds a CPS knowledge graph $\mathcal{G}=\langle I, V, L, G\rangle$ to capture the knowledge encoded in the CPS documents. Then, the PSL program is used to resolve the semantic heterogeneity conflicts existing among the entities in the different CPS perspectives; a CPS uncertain knowledge graph $\mathcal{G}_u^*=\langle I, V, L, D^*, U^*\rangle$ represents an integrated design of the CPS. Finally, the membership degree threshold is used to select the soft knowledge facts from $\mathcal{G}_u^*$ that in conjunction with the hard knowledge facts in D^* are part of the final integrated design.

Capturing Knowledge Encoded in CPS documents The CPS Knowledge Capture component receives as inputs documents in a given standard containing the description of the perspectives of a CPS design. Next, these documents are automatically transformed into RDF, by following the semantics encoded in the corresponding standard ontology. To this end, a set of mapping rules are executed to create an RDF KG using a standard ontology. Consequently, the output of this component is $\mathcal{G}$, a KG comprising the input data in RDF.

Generating a CPS Uncertain Knowledge Graph The CPS Uncertain KG Generation component creates, based on the input KG, the hard and soft knowledge facts, i.e., the Uncertain KG. To achieve this goal, SemCPS relies on the PSL rules described in Figure 6.9. Next, all facts with degree of membership equal to 1.0 correspond to hard knowledge facts. The rest generated during the evaluation of the rules correspond to soft knowledge facts.

Generating a Final Integrated CPS Design The Final Integrated CPS Design Generation component utilizes a membership degree threshold to select the facts in the CPS uncertain KG.

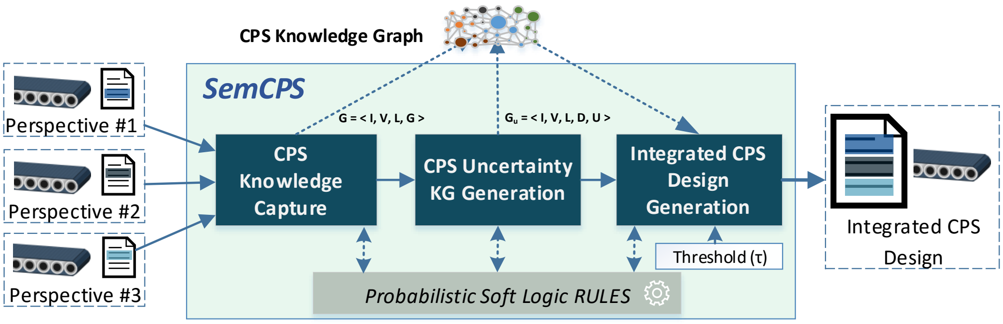


Figure 6.9: **The SemCPS Architecture**. SemCPS receives documents describing a CPS from various perspectives; they are represented in standards like AML. SemCPS outputs a final design document describing the integration of the perspectives, i.e., a knowledge graph. (1) Input documents are represented as an RDF knowledge graph. (2) A rule-based system is used to identify semantic heterogeneity conflicts among the perspectives represented in a knowledge graph. (3) A rule-based system is utilized to resolve semantic heterogeneity conflicts and produced the final integrated CPS design.

Facts with scores below the value of the threshold are removed. Next, the rest of the facts are joined by applying a **Union** policy. The joined facts are part of the final integrated design.

6.5 Empirical Evaluation

We empirically study the effectiveness of the presented approaches in the solution of the problem of CPS conflict identification and CPS conflict resolution. The goal of the experiment is to analyze the impact of: (1) the number of semantic heterogeneity conflicts on the effectiveness of the approaches; and (2) the size of CPS perspectives on the efficiency of the approaches. Particularly, we assess the following research questions:

RQ1) Does the type of heterogeneity conflict between the perspectives of a CPS impact on the effectiveness of the compared approaches?

RQ2) Does the size of the perspectives of a CPS affect the effectiveness of the compared approaches?

RQ3) Does the degree of membership threshold impact on the effectiveness of SemCPS?

We compare the SemCPS and Alligator approaches with EDOAL and SILK. With the goal to compare the approaches, rules in EDOAL and SILK are created. The aim of these rules is to resolve semantic heterogeneity problems between CPS perspectives.[4] For EDOAL and SILK, SPARQL queries are generated based on their rules. These queries are then executed on top of the CPS perspectives after their conversion to RDF.

To the best of our knowledge, real-world publicly benchmarks in the industry domain are not available. Moreover, many of the smart manufacturing standards are not even publicly accessible. This complicates more the possibility to access to a full benchmark of real-world CPS documents. To address this conflict a generator of CPS perspectives is implemented. The

[4] https://github.com/i40-Tools/Related-Integration-Tools

Table 6.1: **Experiment 1: the effectiveness of the compared approaches based on different types of semantic heterogeneity conflicts.** The effectiveness of SemCPS, Alligator, EDOAL, and SILK is compared in terms of precision (**P**), recall (**R**) and F-measure (**F1**). In general, SemCPS exhibits better effectiveness for the increasing number of semantic heterogeneity conflicts than Alligator, EDOAL, and SILK.

Conflicts	**SemCPS**			**Alligator**			**EDOAL**			**SILK**		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
M1	1.0	0.94	**0.96**	1.0	0.93	**0.96**	0.92	0.37	0.53	0.92	0.37	0.53
M1 - M2	0.98	0.99	**0.99**	1.0	0.95	0.98	0.76	0.61	0.67	0.72	0.61	0.66
M1 - M3	0.94	0.85	**0.9**	0.92	0.72	0.81	0.9	0.65	0.75	0.86	0.65	0.74
M1 - M4	0.89	0.91	**0.9**	0.91	0.61	0.73	0.77	0.63	0.69	0.77	0.63	0.69
M1 - M5	0.81	0.87	0.84	1.0	0.81	**0.89**	0.78	0.59	0.67	0.6	0.59	0.59
M1 - M6	0.89	0.87	**0.88**	0.95	0.72	0.82	0.8	0.77	0.79	0.6	0.77	0.68
M1 - M7	0.91	0.83	**0.87**	0.95	0.82	0.71	0.7	0.77	0.73	0.54	0.77	0.64

Table 6.2: **Experiment 2: The effectiveness of the compared approaches based on the size of CPS perspectives.** The effectiveness of SemCPS, Alligator, EDOAL, and SILK is compared in terms of precision (**P**), recall (**R**) and F-measure (**F1**). SemCPS exhibits better effectiveness for the increasing number of entities in perspectives than the rest of the approaches.

Entities	**SemCPS**			**Alligator**			**EDOAL**			**SILK**		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
30	0.87	1.0	**0.93**	1.0	0.76	0.88	1.0	0.84	0.91	1.0	0.84	0.91
60	0.99	0.98	**0.99**	1.0	0.73	0.84	0.75	0.91	0.82	0.95	0.59	0.9
180	0.9	1.0	**0.95**	0.95	0.72	0.83	0.88	0.91	0.88	0.86	0.91	0.89
210	0.89	0.92	**0.9**	0.97	0.80	0.87	0.61	0.97	0.75	0.73	0.97	0.83
600	0.78	1.0	**0.87**	0.93	0.79	0.85	0.87	0.77	0.81	0.87	0.77	0.81

generator creates a testbed of CPS perspectives representing real-world scenarios and allow for the empirical evaluation of SemCPS.

6.5.1 CPS Document Generator

We develop a generator able to produce different perspectives of a seed real-world CPS.[5] Testbeds with 70 seed CPS and two perspectives per CPS are considered as input to the generator. The structure of real-world AML documents[6] is investigated to manually build a CPS seed. Based on this CPS seed, two perspectives comprising semantic interoperability conflicts are created. We handcrafted these three documents, i.e., a seed CPS and the two perspectives for each seed CPS. For each type of semantic heterogeneity conflicts, i.e., from **M1** and **M7**, ten testbeds are created, thus 70 seed CPS and two perspectives per CPS are developed.

Based on a *Poisson* distribution, a value between one and seven is selected; it simulates the number of semantic heterogeneity conflicts that exist in each perspective. The parameter λ of the *Poisson* distribution indicates the average number of heterogeneity among perspectives; λ is set to two and simulates an average of 16 heterogeneity pair-wise perspectives. Thus, generated

[5] https://github.com/i40-Tools/CPSDocumentGenerator
[6] https://raw.githubusercontent.com/i40-Tools/iafCaseStudy/master/IAF_AMLModel_journal.aml

Table 6.3: **Testbed Description**. Minimal and maximal configurations (Config.) in terms of number of entities, relations, heterogeneity, and document size

Config.	# entities	# Relations	# M1-M7	Size (KB)
Minimal	20	8	1	5.7
Maximal	600	350	7	116.2

perspectives include components, attributes, and relations which are commonly included in real-world AML documents.

6.5.2 Experiment Configuration

Testbeds. Each perspective has in average 200 entities related using 100 relations; furthermore, in average three interoperability conflicts occur between the two perspectives of a CPS. Table 6.3 summarizes the features of the evaluated CPS perspectives. As Table 6.3 shows, the testbed comprises variety of entities, relations, and heterogeneity conflicts with the aim of simulating real-world CPS designs.

Gold Standard. Formally, the Gold Standard corresponds to an ideal conflict-free CPS knowledge graph document $\mathcal{G}^*$, for each pair of the CPS documents in the testbeds. The Gold Standard includes CPS knowledge graphs–$\mathcal{G}$–corresponding to complete integrated designs of CPS perspectives in the testbed.
Metrics. We evaluate ALLIGATOR and SEMCPS in terms of the following metrics:

a) **Precision** is the fraction of the semantic interoperability conflicts, i.e., $SC(\mathcal{G})$, that are semantic interoperability conflicts in a CPS document, i.e., conflicts$(\mathcal{G} \mid \mathcal{G}^*, \sigma)$.

$$Precision = \frac{|SC(\mathcal{G}) \cap conflicts(\mathcal{G} \mid \mathcal{G}^*, \sigma)|}{|SC(\mathcal{G})|}$$

a) **Recall** is the fraction of the semantic interoperability conflicts that are in an CPS document, i.e., conflicts$(\mathcal{G} \mid \mathcal{G}^*, \sigma)$.

$$Recall = \frac{|SC(\mathcal{G}) \cap conflicts(\mathcal{G} \mid \mathcal{G}^*, \sigma)|}{|conflicts(\mathcal{G} \mid \mathcal{G}^*, \sigma)|}$$

a) **F-Measure (F1)** is the harmonic mean of precision and recall.

Implementation. Experiments are run on a Windows 8 machine with an Intel I7-4710HQ 2.5 GHz CPU and 16 GB 1333 MHz DDR3 RAM. Both approaches are implemented in Java 1.8. A CPS extraction module is developed as a part of Krextor to transform CPS documents described in AML into RDF graphs. This module comprises a set of mapping rules[7] that are executed in Krextor to create RDF graphs out of AML documents using the AML ontology. Results can be reproduced by using the generator along with data for the experiments[8]; The

[7] https://raw.githubusercontent.com/EIS-Bonn/krextor/master/src/xslt/extract/aml.xsl
[8] https://github.com/i40-Tools/HeterogeneityExampleData/tree/master/AutomationML

ALLIGATOR[9] and SEMCPS[10] frameworks are also available in Github. For ALLIGATOR we implemented the *Deductive System Engine* as a meta-interpreter in Prolog that follows the semi-naïve bottom-up evaluation of Datalog programs [224]; we utilized SWI-Prolog version 7.2.3 and the Prolog Development Tool (PDT).[11] ALLIGATOR uses the CPS extraction module to transform of the RDF files into Datalog extensional predicates is implemented in Java 1.8. SEMCPS is implemented in PSL 1.2.1. SEMCPS also employs the CPS extraction module to transform the perspectives files in AML into the RDF files. Then, it utilizes the RDF files to transform them into the PSL representation.

Impact of the Type of Semantic Interoperability Conflict In order to answer **RQ1**, 70 pairs of CPS perspectives are considered. During an iteration i where $1< i <7$, the two perspectives of each of the 70 CPS have only heterogeneity conflicts of type $\mathbf{M}_i$; Table 6.1 reports on the effectiveness the compared approaches for each iteration in terms of the average of precision, recall and F1. These observed results (cf. Table 6.1) suggest that the behavior of the approaches is little affected by the increasing type of heterogeneity conflicts. For this experiment, the membership degree threshold is set to 0.5 in SEMCPS. Overall, SEMCPS exhibits better results in recall and F1 than ALLIGATOR, EDOAL, and SILK. However, in some cases, ALLIGATOR shows better values of precision.

Impact of the Size of CPS Perspectives. To assess **RQ2**, sizes of the two perspectives of a seed CPS are changed; the experiment is run in five iterations. In iteration one, 30 entities are included in each perspective; then 60, 180, 210 and 600 entities are considered during the following iterations, respectively. For this experiment, the membership degree threshold is set to 0.5 in SEMCPS. ALLIGATOR, EDOAL, and SILK are executed on top of the same CPS perspectives documents. Table 6.2 reports on the effectiveness of SEMCPS, ALLIGATOR,

9 https://github.com/i40-Tools/Alligator
10 https://github.com/i40-Tools/SemCPS
11 https://sewiki.iai.uni-bonn.de/research/pdt/docs/start

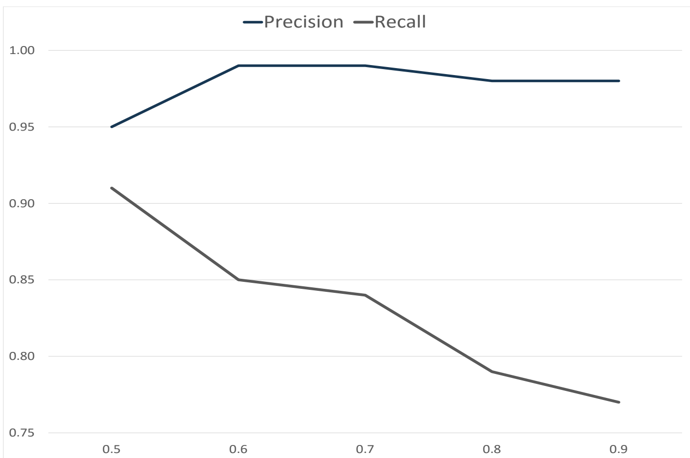
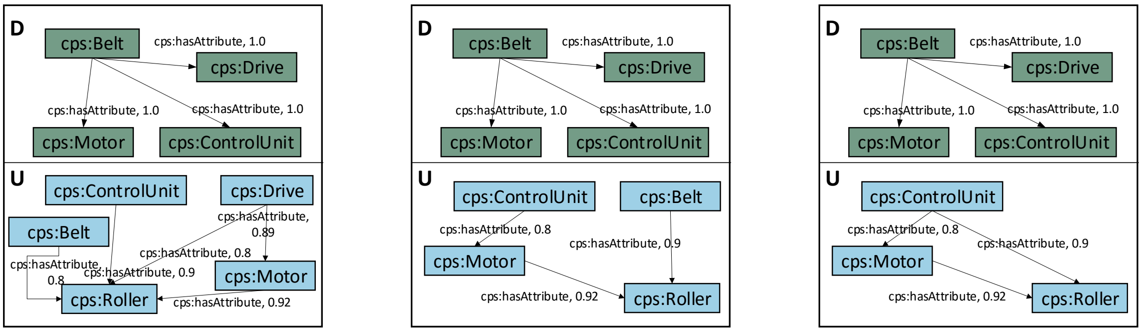


Figure 6.10: **Membership degree Threshold.** Values of precision and recall are shown with different values of the membership degree threshold, i.e., from 0.5 to 0.9. Values of precision are not affected whereas recall decreases up to approximately 0.75 in the last threshold of 0.9.



(a) $\mathcal{G}_u1$ (b) $\mathcal{G}_u2$ (c) $\mathcal{G}_u3$

Figure 6.11: **Uncertain KGs for alternatives of CPS final design**. Uncertain KGs are built based on the alternatives of the motivating example (cf. Figure 6.4). They combine hard (D) and soft(U) knowledge facts; (a), (b), and (c) represent alternate integrated designs.

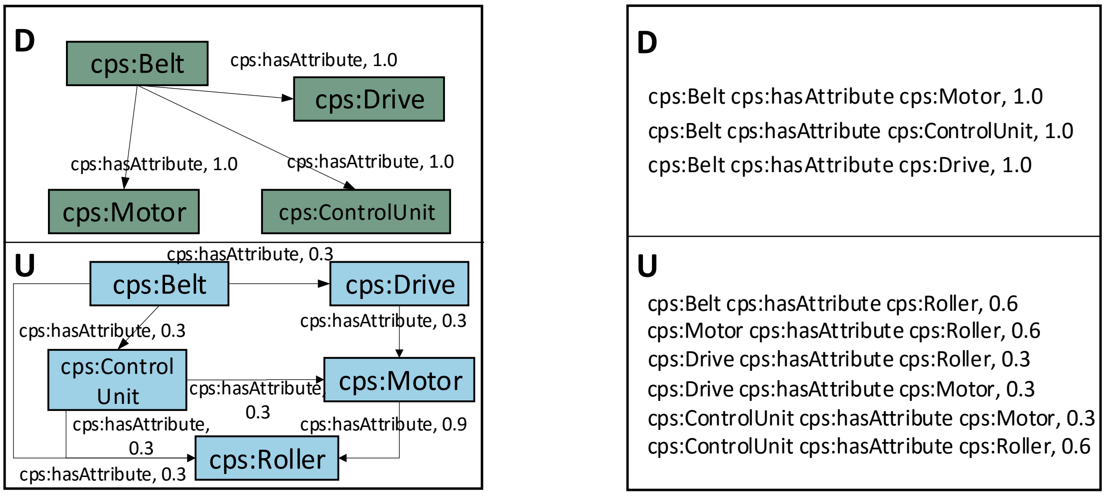
EDOAL, and SILK in terms of the average of precision, recall and F1. In general, SemCPS performs better than the compared frameworks. Also in this experiment, for some cases, the precision values of Alligator are better than the rest.

Impact of the Degree of the Membership Threshold. To evaluate **RQ3**, SemCPS is executed five times with a variation in the membership degree threshold from 0.5 up to 0.9. The execution is done with 210 entities for each perspective. As shown in Figure 6.10, precision is not affected whereas recall decreases up to approximately 0.75 in the last threshold, i.e., 0.9. The membership degree threshold has lowered the effectiveness since in every execution where the threshold is incremented, more soft knowledge facts are excluded from the final integrated design $\mathcal{G}^*$; thus lowering recall. These results suggest that the membership degree threshold impacts on the effectiveness of SemCPS. High values of the membership degree threshold may imply the elimination of soft knowledge facts that are part of the final integrated design. As a consequence, recall is negatively impacted.

6.6 Discussion

In this section, we examine the behavior of the proposed approaches when considering uncertainty in CPS design. For this purpose, we delve into the two cases of the motivating example. Firstly, we investigate the approaches without contemplating uncertainty, i.e., the presence of only hard knowledge facts (cf. Figure 6.3). Secondly, uncertainty is taking into account, i.e., the combination of hard and soft knowledge facts (cf. Figure 6.4).

Listing 6.10 shows the rules that Alligator employs to identify and resolve the conflicts of the two cases of the motivating example (cf. Figure 6.3(a) and Figure 6.13(a)). On the one hand, the rationale behind these types of rules can be read as follows: if two attributes, i.e., $E1$ and $E2$ have a relation with a common attribute, i.e., $E3$, then they are related to each other. When applying the Alligator rules to the motivating example, they are capable to identify the relations between the entities as well as resolve the interoperability conflicts. However, the rules presented in the Alligator approach provide a unique solution



(a) Ideal uncertain KG $\mathcal{G}_u$ (b) $\mathcal{G}_u$ soft and hard facts

Figure 6.12: **Ideal uncertain KG G_u and the associated hard (D) and soft (U) knowledge facts**. The G_u comprises the set of hard and soft knowledge facts. G_u maximizes the probability of the alternate designs by considering the soft knowledge facts of highest probability. Thus, the G_u can be seen as a complete integrated design of the CPS.

to the problem, i.e., they are able to compute only the hard knowledge facts. These types of rules do not consider the uncertainty, i.e., soft knowledge facts. Soft knowledge facts are typical for the description of CPS as illustrated in the motivating example (cf. Figure 6.4(b)).

```
hasAttribute(E1,E2) ⇒ isRelated(E1,E2)
hasAttribute(E1,E3) ∧ hasAttribute(E2,E3) ⇒ isRelated(E1,E2)
isRelated(E1,E2) ∧ isRelated(E2,E3) ⇒ isRelated(E1,E3)
```

Listing 6.10: **Alligator Rules for determining relations in the motivating example**

On the other hand, in the case of SemCPS, the rules can be read as follows: if two attributes, i.e., $E1$ and $E2$ have a relation with a common attribute, i.e., $E3$, then they are *probably* related to each other. Further, these rules have a value, e.g., 0.9. This value represents the importance of a given rule for the domain under study. Listing 6.11 depicts the rules that uses SemCPS to identify and resolve the conflicts of the motivating example (cf. Figure 6.13(a)). SemCPS relies on these rules for identifying all the relations between the entities. Moreover, SemCPS computes possible alternate designs and obtain, out of the alternate designs, the most probable one. It is important to note that the most probable design corresponds to the final integrated design of a CPS.

```
hasAttribute(E1,E2) ∧ ⇒ isRelated(E1,E2) | 0.4
hasAttribute(E1,E3) ∧ hasAttribute(E2, E3) ∧ (E1 != E2)  ⇒ isRelated(E1,E2) | 0.9
isRelated(E1,E2) ∧ isRelated(E2, E3) ∧ (E1 != E3)  ⇒ isRelated(E1,E2) | 0.8
```

Listing 6.11: **SemCPS Rules for determining relations in the motivating example**

No Uncertainty We first analyze the case when no uncertainty in CPS design is considered by the approaches. This implies the presence of only hard knowledge facts between the CPS entities.

```
Belt  hasAttribute  Motor
Belt  hasAttribute  ControlUnit
Belt  hasAttribute  Roller
Belt  hasAttribute  Drive
Motor hasAttribute  Roller
```

(a) ALLIGATOR–hard knowledge facts

```
Belt  hasAttribute  Motor        1.0
Belt  hasAttribute  ControlUnit  1.0
Belt  hasAttribute  Roller       1.0
Belt  hasAttribute  Drive        1.0
Motor hasAttribute  Roller       1.0
```

(b) SEMCPS–hard knowledge facts

Figure 6.13: **Results of the Alligator and SemCPS approaches with only hard knowledge facts**. The results after applying the rules defined in the ALLIGATOR and SEMCPS to the example defined in Figure 6.3. (a) ALLIGATOR computed facts. (b) SEMCPS computed facts. Both approaches compute the same relations between the entities when only hard knowledge facts exist in the CPS design.

We executed ALLIGATOR and SEMCPS on top of the first case of the presented example when no uncertainty is considered (cf. Figure 6.3). The results of the behavior of the approaches with only hard knowledge facts are illustrated in Figure 6.13. As can be seen from the Figure, the same number of relations are computed by ALLIGATOR and SEMCPS. This means that the two approaches are capable to determine the alternative presented in Figure 6.3(b) as the final integrated design.

Considering Uncertainty We then investigate the case in which the approaches consider the existing uncertainty in CPS design. The uncertainty is indicated by the presence of hard and soft knowledge facts between the entities. Figure 6.14 depicts the results of the two approaches in this case. As can be interpreted from the Figure, SEMCPS is capable of finding more relations than ALLIGATOR. SEMCPS considers the existence of relations that are likely to appear, e.g., *Motor hasAttribute Roller* and *Drive hasAttribute Motor*. This enables to contemplate all possible relations between the entities to create possible alternatives. Furthermore, SEMCPS computes the most probable alternative which matches with the final CPS design. Figure 6.4(b) further explains this example. The Figure shows possible alternate integrated CPS designs when applying SEMCPS rules to the example depicted in Figure 6.13(a). In *Alternative1*, all the entities and relations from the three given perspectives are included: `Belt` is related to `Drive`, `Control Unit`, and `Motor`. In addition, `Roller` receives relations from `Motor`, `Control Unit`, and `Drive`. The granularity description of `Belt` is *compatible* with the software and electrical perspectives, while the entities present in all the perspectives are preserved. On the other hand, neither *Alternative2* or *Alternative3* includes all the relations between the entities. Thus, the *Alternative1* seems to be the most *complete* according to the specifications of this CPS design; however, uncertainty about the membership of entities like `Drive` and `Control Unit` requires to be modeled. As a result, if such membership is not contemplated, possible relations are to be missing in the final integrated design. Hence, important knowledge for the resolution of semantic interoperability conflicts in CPS design is lacking. On the contrary, by exploring the probable relations between the entities, this knowledge is contemplated for the resolution of the conflicts.

In addition to the existence of uncertainty, we then analyze the conditions that favor the effectiveness of the ALLIGATOR or SEMCPS approach. This analysis is based on the results of the performed evaluation. Overall, the effectiveness of SEMCPS outperforms ALLIGATOR with respect to the values of F-Measure in the first two experiments. For example, in the second experiment, i.e., size of CPS perspectives, for all the cases SEMCPS performs better than ALLIGATOR (cf. Table 6.2). However, in the first experiment, i.e., different types of semantic heterogeneities, there exist two cases in which ALLIGATOR has equal or better results

```
Belt         hasAttribute  Drive
Belt         hasAttribute  ControlUnit2
Belt         hasAttribute  Motor
Belt         hasAttribute  Roller
ControlUnit  hasAttribute  Roller
ControlUnit  hasAttribute  Motor
Drive        hasAttribute  Roller
```

(a) ALLIGATOR–hard and soft knowledge facts

```
Belt         hasAttribute  Drive         1.0
Belt         hasAttribute  ControlUnit2  1.0
Belt         hasAttribute  Motor         1.0
Belt         hasAttribute  Roller        0.8
ControlUnit  hasAttribute  Roller        0.9
ControlUnit  hasAttribute  Motor         0.9
Drive        hasAttribute  Roller        0.93
Motor        hasAttribute  Roller        0.92
Drive        hasAttribute  Motor         0.89
```

(b) SEMCPS–hard and soft knowledge facts

Figure 6.14: **Results of the Alligator and SemCPS approaches considering hard and soft knowledge facts**. The result after applying the rules defined in the ALLIGATOR and SEMCPS approach to the example defined in Figure 6.4. (a) ALLIGATOR computed facts. (b) SEMCPS computed facts. SEMCPS is able to compute more relations than ALLIGATOR. By relying on SEMCPS the most probable alternative can be computed. The most probable alternative corresponds to the final integrated design when hard and soft knowledge facts are present.

than SEMCPS (cf. Table 6.1). First, the occurrence of **M1** wherein both approaches perform equally. Also, there is a case in which ALLIGATOR exhibits better results that SEMCPS, i.e., the combination of **M1** up to **M5**. The reason for this seems to be that the combination of these specific conflicts, along with the number of hard and soft knowledge facts, favor the better results of ALLIGATOR.

6.7 Concluding Remarks

In this chapter, we tackle the problem of integrating CPS into knowledge graphs by applying Deductive Databases and Probabilistic Soft Logic approaches. We implemented these formalizations in ALLIGATOR and SEMCPS, respectively. First, we presented ALLIGATOR, a deductive approach for the identification and solution of semantic interoperability conflicts between CPS documents. ALLIGATOR encodes the knowledge of the CPS perspectives in a knowledge graph. Then, ALLIGATOR relies on the Datalog rules to represent the knowledge that characterizes different types of semantic heterogeneity conflicts in CPS documents. The set of Datalog rules are utilized to identify and resolve the semantic interoperability conflicts among CPS perspectives. Second, we introduced SEMCPS, an approach for enabling the integration of CPS descriptions into knowledge graphs. SEMCPS uses Probabilistic Soft Logic to capture the knowledge that characterizes different types of semantic heterogeneity in CPS documents. As part of the SEMCPS approach, we defined the concept of uncertain knowledge graphs. Uncertain knowledge graphs are capable to represent the uncertainty, which is typical in CPS design. Uncertain knowledge graphs comprise hard and soft knowledge facts for representing the entities of the CPS perspectives.

We empirically evaluated the presented approach against a testbed of AML document representing CPS perspectives to be integrated. Existing approaches such as EDOAL and SILK are also considered in this evaluation. The results of the empirical evaluation indicate that ALLIGATOR is able to effectively resolve the problems of *CPS Conflict Identification* and *CPS Conflict Resolution*. Results of the empirical evaluation also suggest that SEMCPS is able to effectively resolve the problem of integrating CPS perspectives by using *Uncertain Knowledge Graphs* of I40 related standards such as AML. In general, SEMCPS exhibits better performance than Alligator, EDOAL, and SILK when executed with an accumulative types of semantic heterogeneity conflicts and when an increasing number of entities are added. However, is certain cases Alligator outperforms the compared approaches with regard to the precision. The effectiveness of SemCPS is impacted by higher values of the membership degree threshold. The reason for this is that in every execution where the threshold is incremented, more soft knowledge facts are excluded from the final integrated design. Therefore, the recall is lowered.

We discuss the behavior of the proposed approaches when considering uncertainty in CPS design. To this aim, an example of CPS design is examined in this work. Based on the observed results, we can conclude that there exist conditions that favor the behavior of the approaches. In the case that no uncertainty is present between the CPS perspectives, i.e., only hard knowledge facts, Alligator seems to be a better choice. However, in cases that uncertainty is present, SemCPS enables the identification of the most probable design. The most probable design matches with the final integrated design of a CPS. By automatizing a crucial part of the engineering and modeling processes, Alligator and SemCPS address a key pillar of the I40 vision. Although the initial implementation and evaluation of the approaches is focused on AML, it can be easily transferable to other I40 standards.

CHAPTER 7

Applications of Semantic Data Integration to Industry 4.0 Scenarios

Although the vision of digitizing production and manufacturing has gained much traction lately, it is still not clear how it can actually be *implemented* in an interoperable way using concrete standards and technologies [228]. A key challenge is to enable industrial devices to communicate and to *understand* each other as a prerequisite for cooperation scenarios [168].

Integrating all relevant information and automating as many production steps as possible is the central goal of the I40 vision [229]. Instead of envisioning one monolithic system or database, a decentral semantic integration is pursued, i.e., the formal description and linking of all relevant assets and data sources based on an aligned set of ontologies and their data – *a knowledge graph*. This allows for structured querying and analysis across individual assets and data sources while resolving semantic interoperability conflicts in the domain.

Problem statement. In this chapter, we investigate the applicability of the knowledge graph-based approach for integrating data. For this purpose, we report on a case study in which such a knowledge graph is proposed. This chapter addresses the application component in the general proposed solutions. In particular, the general application component is focused on the Industry 4.0 standards layer (cf. Figure 1.2).

The following research question is investigated in this chapter:

> **RQ4**: How can a knowledge graph-based integration of entities be applied in Industry 4.0 real-world scenarios?

Proposed solution. We address the problem of integrating data from different data sources in a manufacturing company by applying a knowledge graph-based approach. The data sources to be integrated, i.e., the **MES**, **BOM**, and **Sensor** can be seen as standards in the I40 domain (cf. Figure 7.1). These data sources are semantically described. Further, the proposed approach integrates the data while enabling the resolution of existing semantic interoperability conflicts. The chapter is based on the following publication [230].[1]

The contributions of this chapter are outlined as follows:

- a knowledge graph-based approach for integrating data in I40 scenarios; the approach

[1] This is a joint work with Niklas Petersen and Lavdim Halilaj both PhD students at the University of Bonn. In this article, I contributed to the development of the knowledge graph, the uses cases, and the analysis of how semantic interoperability conflicts can be resolved with a knowledge graph-based approach.

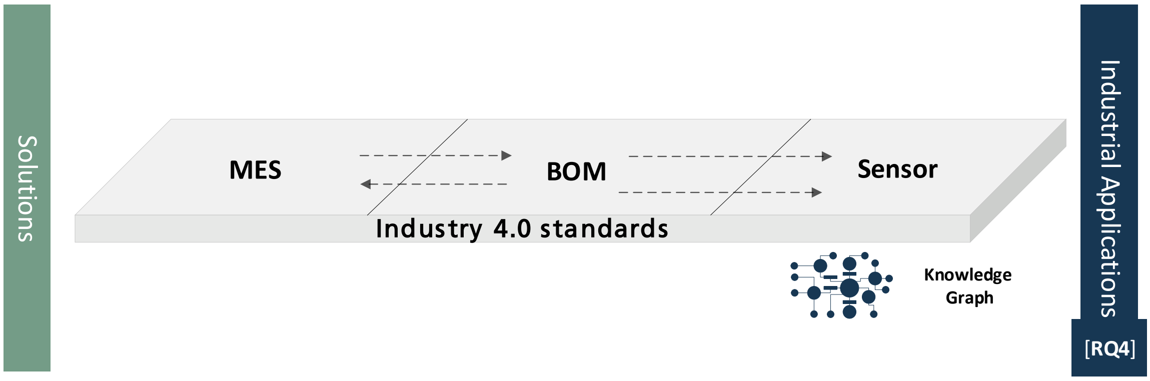


Figure 7.1: **The levels of the problem addressed in this chapter.** We investigated the applicability of a knowledge graph-based approach for semantically integrating data in a real-world case study.

comprises the ontologies, the instance data, the mappings to integrate heterogeneous data sources of the manufacturing company under study while resolving semantic interoperability conflicts between data sources. The semantic integration of data for two typical uses cases in a manufacturing company, i.e., tool availability and energy consumption.

- an architecture depicting a practical solution for semantic data integration based on a knowledge graph approach.

The information model, that acts as the unified model for the knowledge graph, is aligned with important industry standards, such as *RAMI4.0* [20]. The objective of this is to foster data exchange and semantic interoperability in I40 scenarios. The knowledge graph is implemented for a global manufacturing company. We further discuss findings and lessons learned derived from the case study. In Section 7.1, the context, requirements, and motivating scenario are described. The core contributions, i.e., the knowledge graph approach and its implementation, are presented in Section 7.2 and Section 7.3. In Section 7.4, the knowledge graph is applied to the use cases demonstrating its benefits and opportunities. Section 7.5 describes the knowledge graph governance, reports on stakeholder feedback, and summarizes the lessons learned. Finally, concluding remarks for this chapter are presented in Section 7.6.

7.1 Motivating Scenario

The case study of the manufacturing company involved data from distinct sources. While adding new sensors to production lines is straightforward, using the sensor data effectively to improve the production process and decision-making can be cumbersome. Figure 7.2 depicts two excerpts of entities that belong to the data sources of SensorData (**SD**) and Bill of Material (**BOM**). The **SD** data source comprises data about sensors attached to a machine, while the **BOM** contains data of a work order that is to be used for the same machine. On the one hand, in **SD**, the field pointing to the machine uses the word “device”. On the other hand, in the work order entity of **BOM**, the field pointing to the same machine has the name “machine”. While “device” and “machine” are distinct names, they are referring to the same real-world entity, i.e., a machine. This simple example shows a semantic interoperability conflict of **SIC3**. In order to

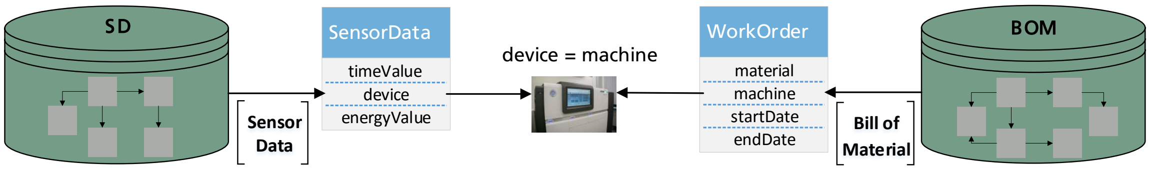


Figure 7.2: **Data sources of Sensor and BOM depicting a semantic interoperability conflict.** In the **SD** data source, the field device of the entity SensorData describes the same real-world entity that the field machine in the WorkOrder entity from the **BOM** data source. A semantic interoperability conflict of **SIC3** exists between the two entities of the data sources. This conflict demands to be resolved for the integration of the existing data.

achieve interoperability in this context, the data required to be integrated. For this purpose, existing semantic interoperability conflicts, e.g., **SIC3**, demand to be resolved when performing the integration.

7.2 Realizing the Knowledge Graph-based Approach

The knowledge graph-based approach aims at a holistic description of the company, its assets and data sources. The objective of the holistic description is to provide an integrated access to all the data of the manufacturing company. This holistic view defines a common data model for the data sources involved where the domain concepts can be expressed at a conceptual level. In addition, for the creation of the knowledge graph, mappings are created. The aim of these mappings is to link the local ontologies representing the data sources, i.e., **SD**, **BOM**, **MES**, with their corresponding schemes.

7.2.1 Methodology

In order to develop the knowledge graph approach, we adapted the methodology proposed in [231]. This methodology provides means for the creation and exploitation of knowledge graphs to integrate data of different sources while resolving semantic interoperability conflicts. For the design of the information model, as well as for the local ontologies, we employed the methodology proposed in chapter 5. First, the purpose and scope of the information model and the local ontologies are defined; then, the domain knowledge is conceptualized and formalized. With the objective to characterize and model semantic interoperability conflicts, the schemas and data of the data sources are inspected. Finally, mappings between the data sources and ontological entities are created. All artifacts are hosted and maintained by VoCol. VoCol supports the requirements of the stakeholders: i) version-control of the ontology; ii) online and offline editing; and iii) support for different ontology editors (by generating a unique serialization before changes are merged to avoid false-positive conflicts [232]).

7.2.2 Purpose and Scope

The knowledge graph comprises: i) a formal description of the physical assets of the company; and ii) formalization of domain-related knowledge of experienced employees about certain tasks and processes within the company. The mappings to database schemas of existing data sources are used to build the knowledge graph. The core concept of the knowledge graph is the *machine*, including its sensor data, usage processes, and human interaction. Therefore, the majority of concepts are defined by their relation to this machine. The scope is set by the motivating examples *tool availability* and *energy consumption* described in Section 7.4.

7.2.3 Capturing Domain Knowledge

The main objective of capturing the domain knowledge is to describe the three data sources of interest, i.e., **MES**, **BOM**, and **SD**. The domain knowledge is captured in different ways:

1. The company provided descriptive material of the domain, descriptions of machines, work orders, processes, as well as sensor data. The types of input material ranged from formatted and unformatted text documents to spreadsheets and SQL dumps.

2. An on-site demonstration of the machine within the factory is given during the project kick-off, including a discussion of further contextual information missing in the material. In subsequent meetings, open questions are clarified and concrete use cases for the knowledge graph are discussed.

3. Well-known ontologies and industry standards are reviewed, intending to build on available domain conceptualizations and formalizations.

4. We created documents to enable easy participation of domain experts by collecting input on the ontology classes and properties in a structured way. These documents are handed over to the domain experts to be reviewed and completed.

All the input information and activities are used to analyze the data of the sources. A special focus is put on the meaning of the data across the data sources. The aim is to understand what data is stored, the meaning of these data with respect to the semantic interoperability conflicts that may exist. When performing the integration, these existing conflicts in the data sources require to be resolved. Therefore, identifying existing semantic interoperability conflicts is of core significance to the approach.

7.2.4 Data Sources

In this section, we describe the data sources of interest for this work: i) **SD**, ii) **BOM**, and iii) the Manufacturing Execution System (**MES**). The **SD** comprises sensor measurements of the machine tool. These measurements record parameters needed for the continuous monitoring of the machine, such as energy, power, temperature, force, and vibration. The **MES** contains information about work orders, shifts, and material numbers. The machine produces assets based on the work order details, which provide the necessary information for the production of a given asset. The **BOM** contains information about the general structure of the company, e.g., work centers, work units, as well as information related to the work orders and the materials needed for a specific production. To achieve interoperability between the data sources, semantic interoperability conflicts may be first identified. To this end, we investigated existing semantic interoperability conflicts and correspondences among their schema.

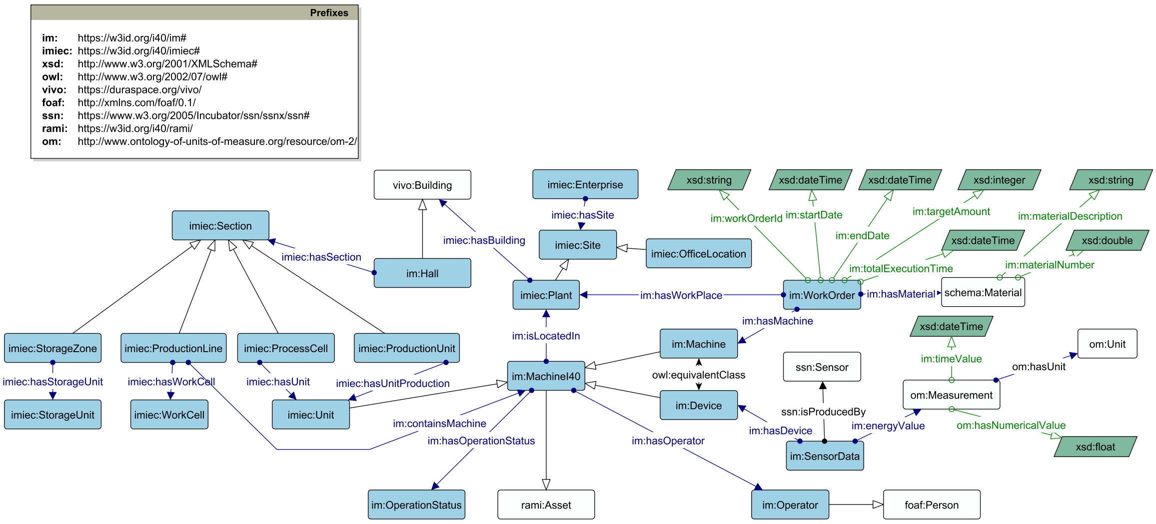


Figure 7.3: **Core concepts of the information model.** Classes and relations of the information model are depicted in blue. Green arrows represent datatype properties. White classes represent reused classes from well-known ontologies such as SSN, schema.org, FOAF, OM, NeoGeo, and VIVO. For the reused properties black arrows are employed. The RAMI4.0 ontology is also utilized for connecting the `im:Machine40` concept with the `rami:Asset`. The information model serves as an unified schema for the semantic annotation of the data sources.

7.2.5 Information Model

The information model outlines main industry concepts and terms that are needed to link the three data sources. Figure 7.3 shows the core concepts of the developed information model which acts as a unified schema for the semantic annotation of the data sources.

Since machines are the main assets of the manufacturing company, they are used as a starting point for the development of the information model. We created the *im:MachineI40* class to model this concept. The `im:MachineI40` is designed as a superclass of the `im:Machine` and `im:Device` classes. Each *im:MachineI40* is located in some *im:Plant*, which is linked to a `vivo:Building`. Each `im:Hall` can contain multiple sections. `im:Plant` and `im:OfficeLocation` are different types of `im:Site`, each one describing distinct locations for the machines. The `im:MachineI40` comprises domain-related properties to describe its `AVO`, i.e., its operation status. Next, it is connected with `im:WorkOrder` to be processed. Each `im:WorkOrder` defines the required `im:Material` and `im:Tool`, as well as which machine should be used by which operator to execute a particular task. In total, the information model comprises 148 classes, 4662 instances, 89 object properties, and 207 datatype properties. We focused on the description of the core concepts that are needed to understand this work.

Reusing well-known ontologies The developed ontologies that are part of the proposed approach, reuse concepts from well-known ontologies. In particular, the concept `vivo:Building`[2] representing physical locations, e.g., a `im:Hall` or a `im:Plant` is employed. To model locations on a map the *NeoGeo*[3] vocabulary is used. Specifically, the concept of `neo:Geometry` for characterizing different types of geometry, e.g., line, polygon, and point. The point concept is utilized to

[2] http://vivoweb.org/
[3] http://geovocab.org/doc/neogeo.html

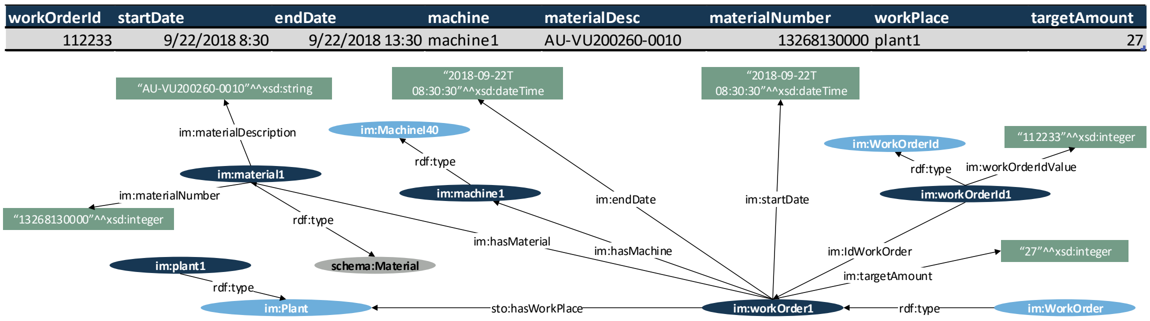


Figure 7.4: **Knowledge Graph Creation example.** An RDF graph is created out of the tabular representation of a work order. The classes and properties of the information model are used to annotate the data of a work order.

describe the latitude, i.e., `geo:lat`, and longitude, i.e., `geo:long` to depict specific locations on a map. The `foaf:Person` class delineates the super concept associated with the person that operates a machine. The `ssn:Sensor` concept as well as the property `ssn:isProducedBy` are reused to represent the sensor data. To encode the semantic of the units of measurements needed in the energy measurement, the OM ontology is employed. The class `schema:Material` is reused to denote the concept of material to be used in a work order. Furthermore, we aligned the information model with ontologies of I40 standards such as RAMI4.0. The goal of this alignment is to consider the `im:MachineI40` concept as an I40 asset, i.e., `rami:Asset`.

7.2.6 Knowledge Graph Creation

The creation of the knowledge graph utilizes the R2RML mappings to transform the data from the data sources to RDF triples.[4] Listing 7.1 describes the R2RML mapping for the work order entity in the **BOM** data source. Further, Figure 7.4 depicts an excerpt of the work order entity. The RDF graph for the work order is obtained out of the tabular representation by utilizing the classes and properties described in the information model. The creation of the knowledge graph relying on the information model resolves the semantic interoperability conflict of **SIC1**.

```
@prefix im:     <https://w3id.org/i40/im#> .
@prefix rr:     <http://www.w3.org/ns/r2rml#> .
@prefix schema: <https://schema.org/> .

<WorkOrderMap>  a  rr:TriplesMap ;
    rr:logicalTable  [ rr:tableName "WorkOrder" ];
    rr:subjectMap    [ rr:template
    "http://.../infomodel/WorkOrder/{WorkOrderId}"; rr:class im:WorkOrder];
    rr:predicateObjectMap
    [ rr:predicate  im:workOrderId;   rr:objectMap [ rr:column  "WorkdOrderId" ]],
    [ rr:predicate  im:startDate;     rr:objectMap [ rr:column  "startDate"]],
    [ rr:predicate  im:endDate;       rr:objectMap [ rr:column  "endDate" ]],
    [ rr:predicate  im:targetAmount;  rr:objectMap [ rr:column  "TargetAmount" ]],
    rr:subjectMap   [ rr:template
    "http://.../infomodel/Tool/{ToolId}"; rr:class  im:Tool ];
    rr:subjectMap   [ rr:template
    "http://.../infomodel/Material/{MaterialId}"; rr:class  schema:Material ]; ...
```

Listing 7.1: **R2RML mapping for the Work Order entity of the BOM data source**

[4] For privacy reasons the real URI is not shown; instead, the best practice of the w3id.org service is used.

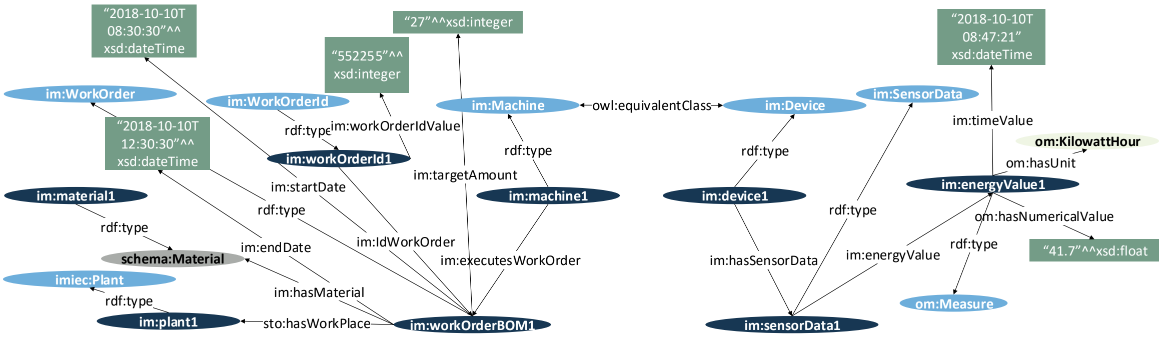


Figure 7.5: **Knowledge Graph Integration example.** An RDF graph representation is created out of the data of a sensor data and work order entities that belong to the **SD** and **BOM** data sources. The same machine is named differently in the two data sources. After transforming the data into a graph representation, the entities are semantically integrated by using the `owl:equivalentClass` property.

Alignment of Concept Identifiers The data sources are inspected to search for the identifiers that are equivalent. For instance, the identifiers from the work order entity that belongs to the **BOM** and **MES** data sources are aligned. When performing this alignment the semantic interoperability conflicts of **SIC2**, and **SIC5** are resolved.

7.2.7 Knowledge Graph Integration

In order to integrate the knowledge of the data sources, semantic interoperability conflicts demand to be resolved. For instance, the same machine is named differently in two of the data sources, i.e., "machine" in the work order entity of the **BOM** and "device" in the energy measurement of **SD**. After transforming the data into a graph representation, entities of the data sources are integrated by using the `owl:equivalentClass` property. Figure 7.5 shows an example of this knowledge graph integration. Additionally, in the **SD**, the energy value is modeled by means of the OM. As explained in Section 5.2.5 this modeling helps to capture the semantics of the units of measurements in I40 scenarios. In this case, the knowledge graph integration and the modeling of the units of measurements enable the resolution of the semantic interoperability conflicts of **SIC2**, **SIC3**, **SIC4**, and **SIC6**.

7.3 Architecture and Implementation

With the objective to provide a uniform interface for accessing heterogeneous distributed data sources, we designed and implemented the architecture illustrated in Figure 7.6. The proposed architecture is extensible and able to accommodate additional components for accessing other types of data sources. The architecture is based on the mediator and wrappers architecture with ontologies as the main artifacts [233]. The mediator enables access to local data sources and conciliates semantic interoperability conflicts between the data sources. In the following, the four main layers of the architecture are described.

The **Application layer** contains client applications that benefit from the unified access interface to the heterogeneous data sources. These applications can be machine agents or human interface applications able to query, explore, analyze and produce human-friendly presentations.

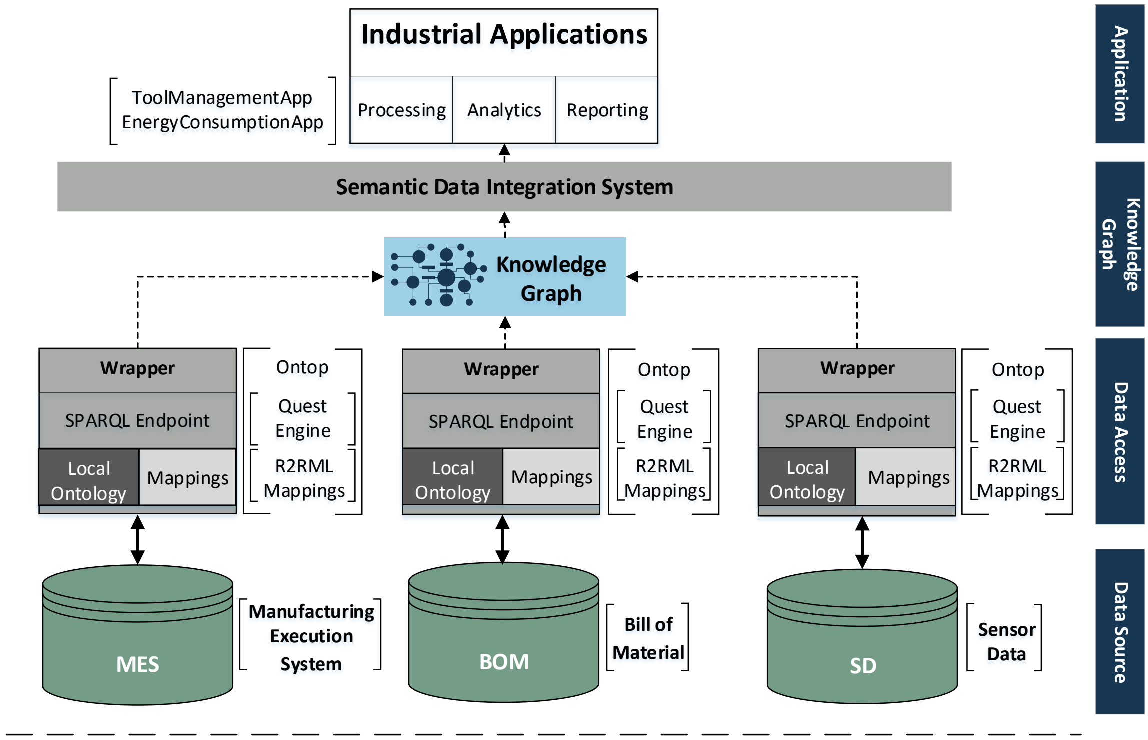


Figure 7.6: **Semantic data integration Architecture.** Different layers of the architecture are presented: i) the data source layer where data sources of the company are placed; ii) the data access layer, which contains the wrappers, the mappings and the access points; iii) the knowledge graph layer comprising ontologies and data; and iv) the application layer which manages client applications to access the integration system [46].

The **Knowledge Graph layer** comprises a mediator that allows for the interoperability of the data sources. This layer encompasses the set of development ontologies along with the data that populates them. In this layer, the meaning of the entities of the domain is expressed by means of the knowledge graph, i.e., the definition of the set of classes, the relations, and instance data. Furthermore, the information model has been created to conceptualize a unified view of the data (cf. Section 7.2). To develop the information model and the local ontologies we followed a *Hybrid Ontology Approach*. This approach is characterized by combining a *Global Ontology Approach* where all data sources are described in an integrated global ontology and the *Multiple Ontology Approach* where separate local ontologies represent the respective data sources [234]. The utilized approach enables new data sources to be added easily, avoiding the need for modifying the mappings or the shared ontology. Accordingly, the developed ontologies are organized in two groups: i) a shared ontology, the information model, to represent the highest level of abstraction of concepts and mappings with external ontologies; and ii) local ontologies representing the schemas of the data sources. This makes the proposed architecture flexible w.r.t. the addition of diverse types of data sources [235]. Reasoning capabilities are executed on top of the knowledge graph to infer new knowledge. The reasoning is executed based on the punctual solutions developed for the use cases, i.e., the use of the `owl:equivalentClass` for mapping concepts and `owl:FunctionalProperty` for mapping properties to the same concept.

The **Data Access layer** consists of various wrappers acting as bridges between client applications and heterogeneous data sources. The wrappers are software components specific for each data source. This layer comprises local ontologies representing a data source. Additionally, this layer provides means to access and to query the local data source. It receives user requests in the form of SPARQL queries, which are translated into the query languages of the respective data sources, and returns the results after query execution. Accessing relational databases is realized using the OBDA paradigm, where ontologies are used as a conceptualization of the unified view of the data, and mappings to connect the defined ontology with the data sources [236]. In particular, the Ontop [145] framework is used to access the data sources, i.e., the **BOM**, **MES**, and **SD** data, which exposes the relational databases as virtual RDF graphs. To executed the SPARQL queries, Quest is utilized. Quest is part of the Ontop framework and provides a SPARQL engine supporting the RDFS and OWL 2 QL entailment regimes inside Ontop. Furthermore, this layer deals with the mappings between the data stored in the data sources and the local ontologies. For the definition of the mappings, we used R2RML. As a result, it is possible to view and access the existing relational databases in the RDF data model.

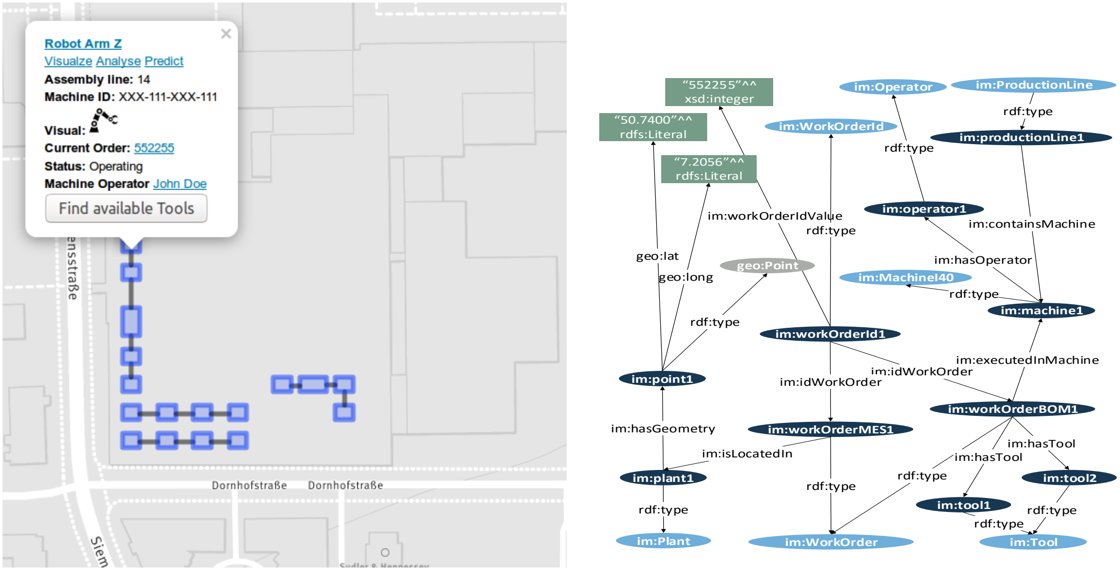
The **Data Source layer** comprises the data sources described in Section 7.2.4. Due to the high dynamicity and the great amount of incoming data, the data sources are replicated and synchronized periodically. Additional types of data sources can be integrated with the overall architecture by defining local ontologies, mappings with the global ontology, and data sources. Furthermore, a wrapper is required for dealing with the specifics requirements of the data of the new source to be added.

7.4 Use Cases

The concrete use cases are based upon a machine newly introduced into the production lines of the company, i.e., a *machine*. This is a machine that requires the mounting of tools to assemble specific metal or rigid products. The new machine features more than 100 embedded sensors that monitor the production. In the following, we applied the developed knowledge graph to the use cases to demonstrate the possibilities resulting from semantically integrated data access.

7.4.1 Tool Availability

In this section, we described the use case of the availability of tools for a given machine. Possible tools to be mounted into the machine are cutters, drillers or polishers. A tool usually consists of multiple parts. The number of parts depends on the manufacturer of the tool, which is not necessarily the same as the manufacturer of the machine. Mounting tools into a machine is a time-consuming task for the machine operator. Uncertain variables of the tools, such as location, availability and utilization rate, play a major role in the efficiency of a work order and of a machine in particular. The production of certain goods may wear a tool out quickly, thus decreasing its overall lifetime and forcing the machine operator to stop the machine and replace it with a new tool. Reducing the idle time for remounting the machine by clearly describing its configuration, location, and weariness, is therefore, one concrete goal to be addressed by the knowledge graph-based approach. In this use case, we asked the following question: where is located the machine and which tools are available in a work order? Figure 7.7(a) depicts a map view of a machine. The sites of the company in which the machine is placed are highlighted based on their geo-location given in the information model. By zooming in, the different locations can be investigated with respect to their functionality, address, on-site buildings up to the level of



(a) Find available Tools (b) Integrated graph data

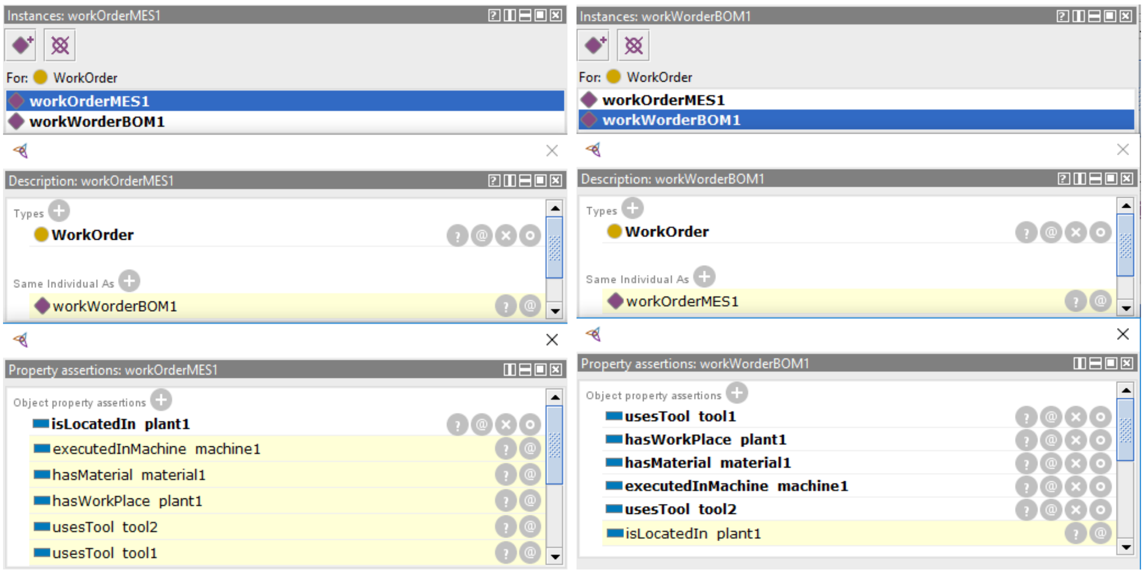
Figure 7.7: **Tool availability of a given machine and the graph representation of the integrated data.** (a) The factory data that can be checked on a map view. The available tools for a given machine can be localized on a map view. (b) The graph representation of the semantically integrated data. To obtain the tool availability of a given machine, data from two data sources, i.e., **MES** and **BOM**, are integrated and semantic interoperability conflicts are resolved.

machines. By clicking on the objects on the map, static and live production data is displayed. With the goal of retrieving this information, data from two of the data sources need to be integrated, i.e., from **MES** and **BOM**.

```
PREFIX im: <https://w3id.org/i40/im#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX neogeo: <http://geovocab.org/doc/neogeo.html>

SELECT ?machineName  ?productionLineR  ?machineId  ?workOrderIdValue  ?operator ?plant  ?tool  ?lat  ?long
WHERE  {
    ?machine         im:machineId  ?machineId ;
                     rdfs:label  ?machineName ;
                     im:hasOperator  ?operator .
    ?productionLine  im:containsMachine  ?machine;
                     im:productionLineId ?productionLineId;
                     rdfs:label ?productionLineName .
    BIND ( CONCAT ( ?productionLineName,  ": ",  ?productionLineId)  AS  ?productionLineR) .
    ?workOrderBOM    im:executedInMachine  ?machine ;
                     im:usesTool  ?tool .
    ?workOrderIdBOM  im:idWorkOrder  ?workOrderBOM ;
                     im:workOrderIdValue  ?workOrderIdValue .
                     im:idWorkOrder  ?workOrderMES .
    ?workOrderMES    im:isLocatedIn  ?plant .
    ?plant           im:hasGeometry  ?point .
                     ?point neogeo:lat   ?lat ;
                            neogeo:long  ?long . }
```

Listing 7.2: **Query to retrieve information about work orders from different data sources**



(a) Work order MES reasoning (b) Work order BOM reasoning

Figure 7.8: **Reasoning of the MES and BOM work orders.** The reasoning is applied on top of the knowledge graph. (a) The `im:workOrderMES1` instance and five new triples, depicted in yellow, are retrieved. (b) The `im:workOrderBOM1` instance and one new triple is obtained. The reasoning based on the `owl:FunctionalProperty` allowed to obtain that the two instances are the equivalent. Moreover, the same set of properties and instances can be integrated into one instance.

By inspecting the data sources we found a matching between the values of the identification of the work orders. In spite the values match, the name of the field that defines the identification of a work order in the two data sources differs. Thus, a semantic interoperability conflict of schema, i.e., **SIC2**, exist. With the aim to solve this conflict, we linked the identification of the work orders by using the functional property `im:idWorkOrder`. A functional property is a property that can have only one value for each instance. Thus, if such property is linked with two instances, i.e., `im:workOrderMES1` and `im:workOrderBOM1`, it means that, when applying the reasoning the two instances are equivalent, i.e., `im:workOrderMES1 owl:sameAs im:workOrderBOM1`. Further, new triples are derived in each one of the entities (cf. Figure 7.8). Since the instances are equivalent the data attached to them can be integrated into one instance. This is required to resolve the existing semantic interoperability conflict (cf. Figure 7.7(b)). The semantic union policy is used here to integrate the data. After the data is integrated into the knowledge graph, the query in Listing 7.2 retrieves the information to show the tool availability, i.e., machine, work orders, locations, and available tools on top of a map (cf. Figure 7.7(a)).

7.4.2 Energy Consumption

In this section, we present the use case of energy consumption. Producing goods with the machine tool is an energy-intensive process. Initially, only the energy costs per factory are known. Sensors are added to track the energy consumption per machine and processed work order. For performing the calculation, data from the added sensors and the work orders, which reside in different data sources, demand to be integrated. Moreover, the semantic interoperability conflicts that occur in the data required to be resolved (cf. Figure 7.2).

Information about energy consumption is critical for the factory to forecast the production process, expenses and maintenance. In the second use case, we examined the following question: what is the *energy consumption* of a machine for a specific time interval of a particular *work order*? To answer this question, data from the **BOM** and **SD** data sources need to be integrated and the existing semantic interoperability conflicts resolved. Listing 7.1 displays an excerpt of the R2RML mappings for the relational database table `WorkOrder`. Among others, it includes the identification of a work order. Also, it consists of the start and end dates, the associated tools, the materials as well as the target production amount.

```
@prefix im: <https://w3id.org/i40/im/#> .
@prefix rr: <http://www.w3.org/ns/r2rml#>
@prefix om: <http://www.ontology-of-units-of-measure.org/resource/om-2/> .

<EnergyConsumptionMap> a rr:TriplesMap;
    rr:logicalTable [rr:tableName "\"View_GetEnergyConsumption\"" ];
    rr:subjectMap    [rr:template "im/{DateOfMeasure}";
    rr:class  im:SensorData ];
    rr:subjectMap    [ rr:template
    "im:energyValue/{EnergyValueId}"; rr:class  om:Measurement];
    rr:predicateObjectMap
    [rr:predicate  im:timeValue;  rr:objectMap        [rr:column  "\"DateOfMeasure\""]],
    [rr:predicate  om:hasNumericalValue; rr:objectMap  [rr:column "\"Value\""]].
```

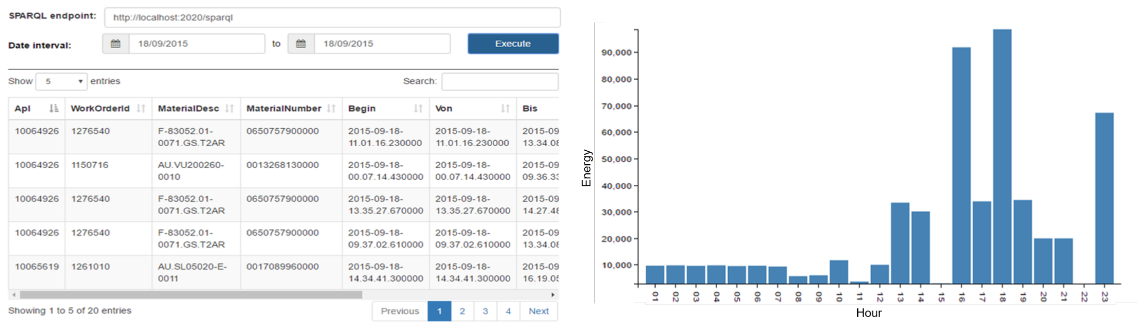
Listing 7.3: **R2RML mapping for energy consumption**

Listing 7.3 shows mappings for the energy consumption. Links to the values of time and measurement of the sensor are created. In order to retrieve the information combining work orders and sensor data, two queries are defined. The first one retrieves information about work orders (cf. Listing 7.4). The second one retrieves the energy consumption values for a work order in a specific time interval (cf. Listing 7.5).

```
PREFIX im: <https://w3id.org/i40/im#>
SELECT  DISTINCT
    ?workOrderId ?beginTime ?dateFrom ?dateTo ?totalExecTime ?targetAmount ?materialDesc ?materialNumber
WHERE  {
    ?workOrder  im:workOrderId     ?workOrderId;
                im:beginWorkOrder  ?beginTime;
                im:startDate       ?dateFrom;
                im:endDate         ?dateTo;
                im:targetAmount    ?targetAmount;
                im:totalExecTime   ?totalExecTime;
                im:hasMaterial     ?material.
    ?material   im:materialNumber  ?materialNumber;
                im:materialDescription  ?materialDesc.
FILTER  (?dateFrom >= dateFrom && ?dateTo <= dateTo)}
```

Listing 7.4: **Query to retrieve information about work orders for a time interval**

This query matches the two concepts named differently in the two data sources by using the `owl:equivalentClass` property. The meaning of using the `owl:equivalentClass` property is that the two connected classes comprise the same set of individuals. The inferences that can be drawn based on this property are as follows: `im:machine1 rdf:type im:Device` and `im:device1 rdf:type im:Machine`. Therefore, the data connected to each one of the instances can be semantically integrated (cf. Figure 7.5). This allowed to obtain the information from the data sources using the knowledge graph and SPARQL queries. Figure 7.9(a) depicts the integrated information of work orders for a given machine, and Figure 7.9(b) shows the energy consumption per hour for that machine for a given day. To finally integrate the data into the knowledge graph we followed the semantic union policy, i.e., creating a new entity with the union



(a) Work order data for a given machine in a time interval of one day

(b) Energy consumption of a given work order within a day

Figure 7.9: **Energy consumption of a work order in a day.** The data of a work order and its energy consumption in the time interval of one day. (a) The data of a work order for a time interval of one day. (b) The energy values for the same time interval. To obtain this information data from the two data sources, i.e., **SD** and **BOM**, are integrated and semantic interoperability conflicts resolved.

of the properties of the matched entities. In this case, the entity `im:machine1` and `im:device1` are unified in one entity (cf. Figure 7.5).

```
PREFIX im: <https://w3id.org/i40/im#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX om: <http://www.wurvoc.org/vocabularies/om-1.8/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT  ?hour  ( (?latest - ?earliest)  AS  ?measurementByHour)  {{
SELECT  ?hour ( MIN (?numValue)  AS  ?earliest) WHERE {
    ?workOrderBOM  im:executedInMachine  ?machine1;
                   im:startDate  ?startDate;
                   im:endDate  ?endDate .
    ?device1       im:hasSensorData  ?sensorData1 .
    ?sensorData1   im:energyValue  ?energyValue .
    ?energyValue   im:timeValue  ?timeValue ;
                   om:hasNumericalValue  ?numValue .
} GROUP BY ( HOURS (?timeValue)  AS  ?hour)}{
SELECT ?hour ( MAX(?numValue)  AS  ?latest) WHERE {
    ?workOrderBOM  im:executedInMachine  ?machine1;
                   im:startDate  ?startDate;
                   im:endDate  ?endDate .
    ?device1       im:hasSensorData  ?sensorData1 .
    ?sensorData1   im:energyValue  ?energyValue .
    ?energyValue   im:timeValue  ?timeValue ;
                   om:hasNumericalValue  ?numValue .
} GROUP BY ( HOURS (?timeValue)  AS  ?hour)}
    ?device1  owl:equivalentClass  ?machine1 .
} ORDER BY  ?hour
```

Listing 7.5: **Query to retrieve the energy consumption of a machine per hour**

7.5 Knowledge Graph Governance

Introducing new technologies is often a challenge for companies. The introduction has to be well-aligned with the organizational structure of the company to balance the *added* value produced for the knowledge graph to the business and the maintenance costs of the technology. Thus, in parallel with the application of the knowledge graph approach, we defined a procedure to support the *governance* of information to ensure the maintenance of the model and uniform decision-making processes. Since the core of the knowledge graph is a network of ontologies with a clear hierarchical and modular structure, there are boards of experts assigned to each part, which are responsible for its maintenance. Decisions cover, for instance, new terms to be included or existing ones to be removed, external vocabularies to be reused and aligned, and the continuous alignment with industry standards implementing the I40 vision, e.g., RAMI4.0, IEC, ISO. Additionally, we provided concrete guidelines for maintaining the knowledge graph along with the use of VoCol.

7.5.1 Evaluation

To gain feedback from the stakeholders involved in the knowledge graph project, we designed a user study. The study comprised a questionnaire that was sent to the stakeholders, asking for anonymous feedback. Table 7.1 lists the questions and results of the study. For evaluating the approach, we are interested in using the feedback from the stakeholders. The feedback is used to evaluate the developed knowledge graph approach as well as semantic technologies in general, based on the experience they gained in the project.

7.5.2 Stakeholder Feedback

To evaluate the knowledge graph and semantic technologies in general, we performed a user study with a questionnaire. Five employees of the manufacturing company, i.e., three IT experts, one analyst, and one consultant, who are actively involved in the project answered the questionnaire. The results varied across the stakeholders: While some regarded the knowledge graph and future potential of semantic technologies as promising, others remained skeptical about its impact within the company. Question 6 asking for the expectations towards semantic technologies (cf. Table 7.1) is answered by nearly all as an "enabler for autonomous systems" and by one as a "potential technology to reduce the number of interfaces". One stakeholder praised the "integration and adaption" capabilities of semantic technologies. Question 7 asking for the biggest bottleneck yielded the following subjective answers: "lack of standardized upper ontologies", "lack of field-proven commercial products", "lack of support for M2M communication standards", "skepticism of the existing IT personnel". While the stakeholders find the advantages of semantic technologies appealing, the lack of ready-to-use business solutions, industrial ontologies, and available IT personnel is halting their efforts to move forward. As a result of the project, the company is actively seeking IT personnel with a background in semantic technologies.

7.5.3 Lessons Learned

Technology awareness within the company After all, the majority of the stakeholders were enthusiastic and committed to developing an integrated knowledge graph and applications on top of it. Nevertheless, reservations on the fitness of the technology and methodology existed from the start. A few stakeholders preferred a bottom-up approach of first gathering and generating internally an overview of the existing schemas and models before involving external parties, such as our research institute. However, the management preferred an *outside view* and put a focus on quick results. Instead of spending time on finding an agreement on how to proceed, speed is the major driving force. Thus, they preferred to try out a "new" technology and methodology, which does not yet have the reputation of strong industrial maturity.

Table 7.1: **Questions and answers of the stakeholders.** A questionnaire with the listed questions is presented to the stakeholders using a Likert scale from 1 to 5. Values of the mean (M) and standard deviation (SD) for each question are outlined.

Question (Likert scale of 1 to 5, 1 = not at all, 5 = very much; M = mean value, SD = standard deviation)	**M**	**SD**
1. Did the developed knowledge graph meet your expectations?	2.4	0.9
2. Do you think investing in semantic technologies can result in a fast ROI?	3.0	1.4
3. Do you consider semantic technologies fit for usage in the manufacturing domain?	3.6	0.9
4. Are you satisfied with the software for semantic technologies available on the market?	2.8	1.3
5. Is it easy to hire personnel with knowledge in semantic technologies?	1.8	0.4
Free-text questions:		
6. What do you expect from semantic technologies in manufacturing contexts?		
7. What is the biggest bottleneck in using semantic technologies in manufacturing contexts?		

Perceived maturity of semantic technologies While semantic technologies are already widely used in some domains, e.g., life sciences, e-commerce or cultural heritage, there is a lack of success stories, technology readiness and show-case applications in most industrial areas. With regard to smaller and innovative products, the penetration of semantic technologies is still relatively small. A typical question when pitching semantic technologies within companies is "Who else in our domain is using them already?". Therefore, it is important to point to successful business projects, even if details on them are usually rare.

Lack of semantic web professionals on the job market Enabling the employees of the manufacturer to extend the knowledge graph by themselves is crucial for the success of the project. Consequently, it is necessary to teach selected stakeholders the relevant concepts and semantic technologies. Hiring new staff experienced with semantic technologies is not necessarily an easy alternative. Compared to relational data management and XML technologies, there is still a gap between the supply of skilled semantic technology staff and the demand of the market.[5]

Importance of knowledge graph governance Of major importance for the company is a clear governance concept around the knowledge graph, answering questions such as who or which department is allowed to access, modify and delete parts of the knowledge graph. An RDF-based knowledge graph has advantages in this regard: i) it enables people across all sites of the company to obtain a holistic view of company data; ii) current data source schemes are enriched with further semantic information, enabling the creation of mappings between similar concepts; and ii) developers can follow a defined and documented process for further evolving and maintaining the knowledge graph.

Building on top of existing systems Accessing data from the existing infrastructure as a virtual RDF graph is an important requirement for the manufacturing company. It avoids the costs of materializing the data into RDF triples and maintaining them redundantly in a triple store, and at the same time, benefits from mature mechanisms for querying, transaction processing, and security of the relational database systems. Three different data access strategies are considered:

[5] For the related field of data science, the European Data Science Academy has conducted extensive studies highlighting such a skill/demand gap all over Europe; cf. Deliverables D1.2 and D1.4 ("Study Evaluation Report 1/2") downloadable from http://edsa-project.eu.

DB in Dumps Relational data to be analyzed is dumped in an isolated place away from the production systems, as not to affect their safety and performance. This strategy is used in cases where the amount of data is small and most likely to be static or updated very rarely.

DB in Replication All data is replicated, allowing direct access from both production systems and new analytic platforms. This solution is considered in cases where data changes frequently and the amount of data is relatively high. It requires allocation of additional resources to achieve a “real-time” synchronization and to avoid performance degradation of the systems in production. We used this strategy to implement our solution since it allows accessing the data sources as a virtual RDF graph and benefit from the maturity of relational database systems.

DB in Production The strategy of accessing data in real-time systems does not require allocating additional resources, such as investment in new hardware or software. Since this strategy exposes a high risk for performance degradation of the real-time systems, whereas sensitive information requires high availability and not providing it on time can have hazardous consequences, we did not apply it in our scenario.

7.6 Concluding Remarks

In this chapter, we investigated the applicability of a knowledge graph-based approach for I40 scenarios. We addressed the problem of integrating data from different data sources in a manufacturing company by applying a knowledge graph-based approach. We viewed the data sources to be integrated as standards in the I40 domain. Thus, this chapter addressed the level of standards in the general architecture of the thesis. Two use cases of core importance for the company are developed, i.e., tool availability and energy consumption. Then, the data sources related to the use cases are semantically described. Existing semantic interoperability conflicts among the data sources are analyzed.

To execute the use cases, we developed a knowledge graph approach for semantically integrating the data of the data sources. The integration of the data contemplated the solution of the semantic interoperability conflicts among the data sources. To achieve this, an architecture for executing the knowledge graph approach is defined. A sets of ontologies are developed to describe the semantics of the data sources. Furthermore, a set of mappings is defined to link the data sources with the ontologies. The architecture enables the integration of data considering the data sources, ontologies, mappings, and applications.

In order to test the proposed solution, a user study is developed. The stakeholders are interviewed with respect to the application of the knowledge graph to the use cases. Additionally, questions with respect to the use of the solution and with the perceived benefits of a knowledge graph-based approaches for manufacturing are presented. In general, the results of the user study demonstrated that the developed solution met the expectations of the stakeholders.

CHAPTER 8

Conclusions and Future Direction

In this thesis, we investigated the problem of enhancing semantic interoperability in Industry 4.0. We proposed a knowledge graph approach for integrating data in these scenarios. The knowledge graph approach enables the integration of data as well as the identification and resolution of semantic interoperability conflicts among Industry 4.0 entities – one of the key challenges in this application domain. The discussion of the research problem, the research questions, as well as the contributions, are presented in Chapter 1. Necessary background concepts are examined in Chapter 2. An overview of state-of-the-art approaches related to the main problem tackled in this thesis is presented in Chapter 3. Then, the subsequent three core chapters of the thesis describe and evaluate three key aspects of the proposed knowledge graph integration approach. Further, a real-world case study, performed in a manufacturing company, is presented in Chapter 7. The case study provides practical evidence regarding the applicability of the knowledge graph approach to the problem of data integration in Industry 4.0 scenarios. Finally in this chapter, the thesis is concluded by revisiting the research questions. To this end, the achieved results are examined in Section 8.1 and some limitations of the work are highlighted in Section 8.2. Section 8.3 outlines possible avenues for future work.

8.1 Revisiting the Research Questions

In order to conduct the work of this thesis, the research problem is divided into four research questions. The objective of the first research question is to investigate whether knowledge graphs are capable to encode the meaning of entities in Industry 4.0 scenarios, particularly those that belong to the specification of standards related to the domain.

> **RQ1**: How can a knowledge graph approach define mappings of standards and standardization frameworks and resolve existing semantic interoperability conflicts among them?

This research question is addressed in Chapter 4. Existing state-of-the-art approaches are limited to classify and relate Industry 4.0 entities without considering the semantics encoded on them. We tackle this by proposing a novel methodology for building knowledge graphs of Industry 4.0 entities. Particularly, this methodology concentrates on semantically describing standards and standardization frameworks. We applied the proposed methodology and built the knowledge graph of Industry 4.0 standards (I40KG). The I40KG comprises semantic descriptions of more than 220 standards, 25 standardization organizations, as well as 10 standardization frameworks. The I40KG semantically describes and annotates standards, as well as relations among them. Furthermore, categorizations of standards with respect to the standardization frameworks are also semantically encoded in the I40KG. These semantics descriptions and annotations help to discover new relations between standards based on existing ones, thus, reducing interoperability conflicts. The knowledge graph internal reasoning step reveals implicit relations among standards. Further, the performed graph analysis is capable to discover most relevant standards, i.e., standards with the largest number of connections in the graph. The knowledge graph interlinking step permits to discover new knowledge about standards and standardization frameworks. We analyze the number of discovered relations among standards and the accuracy of these relations. The observed results indicate that both, the reasoning and linking processes enable to increase the connectivity in the knowledge graph by up to 80%, whilst up to 96% of the relations can be validated. The evidence from this study supports the advantages of a knowledge graph approach for semantically describing and interlinking the knowledge from standards and standardization frameworks.

RQ2: How can knowledge graphs represent semantics encoded in Industry 4.0 entities?

In Chapter 5 this research question is positively answered by demonstrating that knowledge graphs are capable of providing a solid knowledge representation for entities in Industry 4.0 scenarios. We interpret the ontologies as a key part of the knowledge graphs that records the structure, in this case, of the Industry 4.0 domain. In this regard, we proposed a methodology, based on best-practices for ontology building. The methodology is applied for the construction of three ontologies capturing the structure of standards of core importance for Industry 4.0 scenarios. First, the RAMI4.0 model provides a reference architecture for I40 solutions and the Administration Shell concept enables the digital representation of physical assets. Second, the AML ontology covers the AutomationML standard. This standard is crucial in industry solutions for designing CPS from distinct discipline perspectives such as the mechanical, electrical and software engineering ones. Finally, SCORVoc represents the supply chain operations reference model of the APICS industry association. We demonstrate the benefits of the semantic representation of Industry 4.0 entities. Then, knowledge graphs are created for each one of the designed ontologies. Common use cases for the semantic representation in Industry 4.0 scenarios are developed, e.g., the units of measurements. The representation and discovery of semantic interoperability conflicts among entities in these scenarios are introduced. Furthermore, the resolution of conflicts by considering and applying the semantics of the ontologies is developed. The knowledge graph approach for representing and linking entities poses many advantages for the realization of the Industry 4.0 vision. The flexible schema representation, the creation of global unique identifiers for entities, the ease creating a multilingual representation, are some of these advantages that we can observe in the proposed approach.

RQ3: How can existing rule-based approaches be utilized to resolve semantic interoperability conflicts over knowledge graphs?

In Chapter 6, this research question is addressed by proposing Deductive Databases and Probabilistic Soft Logic approaches for creating and exploiting knowledge graphs. We formalize the problem of identifying and resolving conflicts among I40 entities from different CPS perspectives following these two approaches. Knowledge graphs are created for representing the information from different perspectives of CPS design, i.e., mechanical, electrical or software views. We implemented these formalizations in Alligator and SemCPS, respectively. First, we presented Alligator, a deductive approach for the identification and solution of semantic interoperability conflicts between CPS documents. Alligator relies on Datalog to represent knowledge that characterizes different types of semantic interoperability conflicts in CPS documents. Alligator uses a knowledge graph to encode the knowledge of the CPS perspectives. Second, we introduced SemCPS, an approach for enabling the integration of CPS descriptions into knowledge graphs. SemCPS uses Probabilistic Soft Logic to capture the knowledge that characterizes different types of semantic interoperability in CPS documents. As part of the SemCPS approach, we defined the concept of uncertain knowledge graphs. Uncertain knowledge graphs are capable to represent the uncertainty, which is typical in CPS design. Uncertain knowledge graphs comprise hard and soft knowledge facts for representing the entities of the CPS perspectives. An empirical evaluation is performed to compare the proposed approaches with existing ones such as EDOAL and SILK. In general, SemCPS exhibits better performance than Alligator, EDOAL, and SILK when executed with accumulative types of semantic interoperability conflicts and when an increasing number of entities is added. However, in certain cases Alligator outperforms the compared approaches with regard to precision. Furthermore, we analyzed the behavior of Alligator and SemCPS for dealing with uncertainty. In the first case, without considering uncertainty, i.e., only hard knowledge facts, both approaches present similar behavior for identifying and resolving the semantic interoperability conflicts among the perspectives. In the second case, i.e., the combination of hard and soft knowledge facts, the SemCPS approach allowed us to represent the uncertainty which is typical in the CPS design. By relying on this representation, SemCPS is capable to compute many alternatives of CPS design and to choose the most probable one. The most probable alternative matches with the final integrated design for the studied CPS. Taken together, these results suggest that rule-based approaches are capable of identifying and resolving semantic interoperability conflicts in CPS design.

> **RQ4**: How can a knowledge graph-based integration of entities be applied in Industry 4.0 real-world scenarios?

Chapter 7 reports on the results of the semantic-based approach for data integration in real-world industrial scenarios. A case study from an actual manufacturing company is presented and shows the applicability of the knowledge graph approach to integrate heterogeneous data sources in Industry 4.0 scenarios. Two use cases of core importance for the efficiency of factory production are developed, i.e., tool availability and energy consumption. We investigated the data sources of the manufacturing company that are related to the use cases. Existing semantic interoperability conflicts among the data sources are identified and analyzed. Furthermore, to execute the use cases, we applied the knowledge graph approach for resolving the semantic interoperability conflicts. A set of ontologies was developed to describe the semantics of the data sources, i.e., bill of materials, manufacturing execution systems, and sensor data. In addition, a set of mappings are defined to link the data sources to the respective ontologies. An architecture for implementing the knowledge graph approach is defined. The architecture enables the integration of data considering the data sources, ontologies, mappings and applications. The implemented solution is evaluated by interviewing the stakeholders in the manufacturing company. Questions with respect to the use of the solution and the perceived benefits of a knowledge graph approach for manufacturing are presented. We observed that most of the assessment questions received good evaluation results. In conclusion, a knowledge graph approach appears indeed beneficial for integrating data in real-world Industry 4.0 scenarios.

8.2 Limitations

We acknowledge that our research has limitations. The first refers to the breadth of semantic representation of standards. In the development of this thesis, we focus in depth on few standards representing core areas for Industry 4.0. However, we aware there exist a multitude of standards that are employed in the Industry 4.0 domain. Thus, the semantic representation is currently limited to the characteristics of those standards. Nevertheless, the methodology presented in Chapter 5 for modeling ontologies of standards can be also applied to others standards in the domain. In addition, the best practices and lessons learned during the development of the RAMI4.0 ontology, the AutomationML ontology, and the SCORVoc vocabulary, can be extended to other standards in these settings.

The second limitation lies in the lack of benchmarks for entities described in Industry 4.0 related standards. The absence of such benchmarks influenced the results when integrating different entities expressed in Industry 4.0 related standards, e.g., AutomationML. To cope with this limitation we developed a synthetic generator of AutomationML documents representing conflicting perspectives of CPS design.

The third limitation is related to the prototypical and punctual integration with established MES, BOM, and sensor data handling systems. We managed to achieve the semantic data integration for some points. Nevertheless, we identified and described semantic interoperability conflicts which are common in this domain. A knowledge graph comprising data from the data sources involved along with the proposed architecture allowed us to derive a practical solution for the semantic data integration problem in this setting. Further, lessons learned from the interviews and the feedback of the users of the proposed solution provided results of paramount importance for the future development of semantic data integration in the Industry 4.0 domain.

8.3 Future Directions

Utilizing the knowledge encoded in data has become a priority in current times. The use of this knowledge provides added value for processes and services in factories, particularly in Industry 4.0 environments. Industry 4.0 comprises a huge environment with different types of standards and data sources. These standards and data sources demand to be integrated while resolving the existing semantic interoperability conflicts among them. This integration has to respect the meaning of the data. In this section, we examine possible enhancements and envision directions that can be considered for the further development of the problem space tackled by this thesis.

Benchmarks of Knowledge Graphs. The construction and continuous improvement of benchmarks of knowledge graphs representing entities in the Industry 4.0 domain is a necessity for enhancing existing methods of resolving semantic interoperability conflicts. The benchmarks have to consider the levels of semantic interoperability presented in this thesis, i.e., from the standardization framework level up to the CPS level (cf. Figure 1.1). In this regard, it is of particular relevance to take into account benchmarks of Industry 4.0 standards describing documents. For example, a benchmark of the AutomationML and OPC UA standards will be of great benefit for the further development of semantic interoperability in Industry 4.0 scenarios.

Automatically checking Semantic Compatibility of Standards. Typically, there exists a need for combining standards for solving problems in the Industry 4.0 domain. To jointly use the standards, their information models should be checked regarding the semantics of the terms that are defined in those standards. The knowledge of the standards is typically encoded in documents. We envision to use natural language processing techniques to extract this knowledge and codify it in a knowledge graph. This knowledge graph will be connected to the I40KG built in Chapter 4. Then, we aim to use semantic similarity metrics to determine how close are the terms defined by two given standards.

A General Methodology for the Construction and Continuous Refinement of Knowledge Graphs of Industry 4.0 Entities. Since Industry 4.0 is still a rather new concept, there is plenty of room for improvement existing methodologies for achieving semantic data integration in these environments. Knowledge graphs are at the core of this methodology, which is to consider the inner features of Industry 4.0, e.g., the different types of integration needed. The study and characterization of the data sources involved, the identification of semantic interoperability conflicts, the data integration architecture, are some of the key activities to be included in such a methodology. Moreover, the reasoning capabilities of the ontologies to be developed as a part of the knowledge graph is another core area to be investigated. There should be a trade-off between the expressibility of the ontologies covering a specific area and the performance required in that area to meet Industry 4.0 demands.

Knowledge Graphs of Industry 4.0 entities as basis for machine learning analytics. The creation and refinement of knowledge graphs of Industry 4.0 entities will set the basis towards a federated and semantic representations in this domain. These advantages combined with the power of Machine Learning algorithms can provide a vast amount of newly discovered knowledge. Thus, these discovered knowledge is a significant added value for the factories implementing learning techniques on integrated data. Challenges comprise here the explainability of learned patterns for the human assessment. With regard to predictive analytics, machine learning techniques must be responsible, in the sense that violations of the human life and work space (e.g. personal data, machine caused accidents etc.) are avoided as much as possible.

Bibliography

[1] S. Vaidyaa, P. Ambadb and S. Bhoslec, *Industry 4.0–A Glimpse*, Design Engineering **2351** (2018) 9789 (cit. on p. 1).

[2] N. DAgostino, E. Annacondia, A. Bentkus, G. Bianchi, J. Briant, R. Heidel, M. Hoffmeister, P. Lamboley, G. Rossi, M. Mendes, D. S. H. Szigeti, N. Schmidt, P. Tailhades and I. Weber, *The Structure of the Administration Shell: TRILATERAL PERSPECTIVES from France, Italy, and Germany*, tech. rep., Ministry of Economy et al., 2018 (cit. on pp. 1, 13–15).

[3] S. Biffl, A. Lüder and D. Winkler, "Multi-Disciplinary Engineering for Industrie 4.0: Semantic Challenges and Needs", *Semantic Web for Intelligent Engineering Applications*, ed. by S. Bill and M. Sabou, Springer, Springer International Publishing, 2016 17, ISBN: 978-3-319-41490-4 (cit. on p. 2).

[4] K. Thoben, S. Wiesner and T. Wuest, *Industrie 4.0 and Smart Manufacturing - A Review of Research Issues and Application Examples*, IJAT **11** (2017) 4 (cit. on p. 2).

[5] V. Jirkovsky, M. Obitko and V. Marik, *Understanding Data Heterogeneity in the Context of Cyber-Physical Systems Integration*, IEEE Transactions on Industrial Informatics **13** (2017) 660 (cit. on pp. 2, 14, 24, 110).

[6] P. Adolphs, S. Auer, M. Billmann, M. Hankel, R. Heidel, M. Hoffmeister, H. Huhle, M. Jochem, M. Kiele, G. Koschnick, H. Koziolek, L. Linke, R. Pichler, F. Schewe, K. Schneider and B. Waser, *Structure of the Administration Shell*, Status Report, ZVEI and VDI, 2016 (cit. on pp. 2, 15, 25, 52, 74, 75).

[7] S.-W. Lin, B. Miller, J. Durand, G. Bleakley, A. Chigani, R. Martin, B. Murphy and M. Crawford, *The Industrial Internet of Things Volume G1: Reference Architecture*, White Paper IIC:PUB:G1:V1.80:20170131, Industrial Internet Consortium, 2017 (cit. on pp. 2, 16, 17, 25, 26, 50).

[8] S.-W. Lin, B. Murphy, E. Clauer, U. Loewen, R. Neubert, G. Bachmann, M. Pai and M. Hankel, *Reference Architectural Model Industrie 4.0 (RAMI 4.0)*, tech. rep., Industrial Internet Consortium and Plattform Industrie 4.0, 2017, URL: `http://www.iiconsortium.org/pdf/JTG2_Whitepaper_final_20171205.pdf` (cit. on pp. 2, 16, 33, 34, 46, 74).

[9] R. Henssen and M. Schleipen, *Interoperability between OPC UA and AutomationML*, Procedia CIRP **25** (2014) 297, ISSN: 2212-8271 (cit. on pp. 2, 26, 54).

[10] A. S. Thuluva, D. Anicic and S. Rudolph, "Semantic Web of Things for Industry 4.0", *Proceedings of the Doctoral Consortium, Challenge, Industry Track, Tutorials and Posters RuleML, London, UK, July 11-15.* 2017 (cit. on p. 2).

[11] L. Shi, S. Li, X. Yang, J. Qi, G. Pan and B. Zhou, *Semantic health knowledge graph: Semantic integration of heterogeneous medical knowledge and services*, BioMed Research International **2017** (2017) (cit. on pp. 6, 23).

[12] D. Tomic, D. Drenjanac, S. Hoermann and W. Auer, *Experiences with creating a precision dairy farming ontology (DFO) and a knowledge graph for the data integration platform in agriOpenLink*, Agrarinformatika/Journal of Agricultural Informatics **6** (2015) 115 (cit. on pp. 6, 23).

[13] P. A. Szekely, C. A. Knoblock, J. Slepicka, A. Philpot, A. Singh, C. Yin, D. Kapoor, P. Natarajan, D. Marcu, K. Knight, D. Stallard, S. S. Karunamoorthy, R. Bojanapalli, S. Minton, B. Amanatullah, T. Hughes, M. Tamayo, D. Flynt, R. Artiss, S. Chang, T. Chen, G. Hiebel and L. Ferreira, "Building and Using a Knowledge Graph to Combat Human Trafficking", *Proceedings of the 14th International Semantic Web Conference ISWC, Bethlehem, PA, USA, October 11-15*, 2015 205 (cit. on pp. 7, 23).

[14] M. Galkin, S. Auer, M. Vidal and S. Scerri, "Enterprise Knowledge Graphs: A Semantic Approach for Knowledge Management in the Next Generation of Enterprise Information Systems", *Proceedings of the 19th International Conference on Enterprise Information Systems ICEIS, Volume 2, Porto, Portugal, April 26-29*, 2017 88 (cit. on pp. 7, 23).

[15] A. Petrillo, F. D. Felice, R. Cioffi and F. Zomparelli, "Fourth Industrial Revolution: Current Practices, Challenges, and Opportunities", *Digital Transformation in Smart Manufacturing*, InTech, 2018 (cit. on p. 13).

[16] S. Wang, J. Wan, D. Li and C. Zhang, *Implementing Smart Factory of Industrie 4.0: An Outlook*, IJDSN **12** (2016) 3159805:1, URL: https://doi.org/10.1155/2016/3159805 (cit. on p. 14).

[17] S. Potier, "Open standards - Key to success for Industry 4.0", 2014, URL: https://www.slideshare.net/stephane_potier/open-standards-key-to-success-for-industry-40-40582587 (cit. on p. 14).

[18] S. Ali and T. Yue, "U-Test: Evolving, Modelling and Testing Realistic Uncertain Behaviours of Cyber-Physical Systems", *Proceedings of the 8th IEEE International Conference on Software Testing, Verification and Validation, ICST Graz, Austria, April 13-17*, 2015 1 (cit. on p. 14).

[19] J. Lee, B. Bagheri and H.-A. Kao, *A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems*, Manufacturing Letters **3** (2015) 18 (cit. on p. 14).

[20] P. Adolphs, H. Bedenbender, D. Dirzus, M. Ehlich, U. Epple, M. Hankel, R. Heidel, M. Hoffmeister, H. Huhle, B. Kaercher, H. Koziolek, R. Pichler, S. Pollmeier, F. Schewe, A. Walter, B. Waser and M. Wollschlaeger, *Reference Architecture Model Industrie 4.0 (RAMI4.0)*, Status Report, ZVEI and VDI, 2015 (cit. on pp. 15, 16, 46, 50, 75, 132).

[21] S. Lin, B. Miller, J. Durand, R. Joshi, P. Didier, A. Chigani, R. Torenbeek, D. Duggal, R. Martin and G. Bleakley, *Industrial Internet Reference Architecture*, tech. rep., Industrial Internet Consortium (IIC), Tech. Rep, 2015 (cit. on p. 16).

[22] Y. Lu, K. Morris and S. Frechette, *Current standards landscape for smart manufacturing systems*, National Institute of Standards and Technology, NISTIR **8107** (2016) (cit. on pp. 16, 33, 34, 74).

[23] IEC, *IEC 62714-1:2014 Engineering data exchange format for use in industrial automation systems engineering - Automation markup language*, 2014, URL: https://webstore.iec.ch/publication/7388 (cit. on p. 17).

[24] R. Drath, A. Lüder, J. Peschke and L. Hundt, "AutomationML - the glue for seamless automation engineering", *Proceedings of 13th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, September 15-18, 2008, Hamburg, Germany*, 2008 616 (cit. on p. 17).

[25] R. Drath, *Datenaustausch in der Anlagenplanung mit AutomationML: Integration von CAEX, PLCopen XML und COLLADA*, Springer-Verlag, 2009 (cit. on p. 17).

[26] S. Peroni, A. Gangemi and F. Vitali, "Dealing with Markup Semantics", *Proceedings of the 7th International Conference on Semantic Systems*, I-Semantics, ACM, 2011 111, ISBN: 978-1-4503-0621-8 (cit. on pp. 17, 109).

[27] S. Liu, J. Mei, A. Yue and Z. Lin, "XSDL: Making XML Semantics Explicit", *Semantic Web and Databases: 2nd International Workshop, SWDB, Toronto ,Canada, August 29-30, Revised Selected Papers*, ed. by C. Bussler, V. Tannen and I. Fundulaki, Springer Berlin Heidelberg, 2005 64, ISBN: 978-3-540-31839-2 (cit. on pp. 17, 109).

[28] M. Fedai, U. Epple, R. Drath and A. Fay, "A Metamodel for generic data exchange between various CAE Systems", *Proceedings of the 4th mathmod conference*, vol. 24, 2003 1247 (cit. on p. 18).

[29] M. Barnes and E. L. Finch, *COLLADA-Digital Asset Schema Release 1.5. 0 specification*, tech. rep., Khronos Group, Sony Computer Entertainment Inc, 2008 (cit. on p. 18).

[30] S. C. Council, *Supply chain operations reference model*, SCOR, Version **11** (2012) (cit. on pp. 18, 19, 97, 98).

[31] B. Georgise, K.-D. Thoben and F. Marcus Seifert, *Assessing the Existing Performance Measures, and Measurement Systems in Developing Countries: An Ethiopian Study*, Global Journal of Researches In Engineering **13** (2013) (cit. on p. 18).

[32] F. B. Georgise, K.-d. Thoben and M. Seifert, *Implementing the SCOR Model Best Practices for Supply Chain Improvement in Developing Countries*, International Journal of u-and e-Service, Science and Technology **6** (2013) (cit. on p. 18).

[33] F. Lestari, K. Ismail, A. Hamid and W. Sutopo, "Designing supply chain analysis tool using SCOR model (case study in palm oil refinery)", *Proceedings of the IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 2013 (cit. on p. 18).

[34] F. Salazar, M. Caro and J. Cavazos, *Final Review of the Application of the SCOR Model: Supply Chain for Biodiesel Castor–Colombia Case*, Journal of Technology Innovations in Renewable Energy **1** (2012) (cit. on p. 18).

[35] M. Arenas, C. Gutiérrez and J. Pérez, "Foundations of RDF Databases", *Proceedings of the 5th International Summer School of Reasoning Web. Semantic Technologies for Information Systems,Brixen-Bressanone, Italy, Tutorial Lectures*, 2009 158 (cit. on p. 19).

[36] R. Studer, V. R. Benjamins and D. Fensel, *Knowledge Engineering: Principles and Methods*, Data Knowledge Engineering **25** (1998) 161 (cit. on p. 20).

[37] N. Guarino, D. Oberle and S. Staab, "What Is an Ontology?", *Handbook on Ontologies*, 2009 1 (cit. on p. 21).

[38] D. Brickley, R. Guha and B. McBride, *RDF Schema 1.1*, W3C Recommendation, World Wide Web Consortium (W3C), 2014 (cit. on p. 21).

[39] K. Janowicz, P. Hitzler, B. Adams, D. Kolas and C. Vardeman, *Five stars of Linked Data vocabulary use*, Semantic Web **5** (2014) 173 (cit. on p. 23).

[40] C. Bizer, T. Heath and T. Berners-Lee, *Linked Data - The Story So Far*, International Journal on Semantic Web and Information Systems **5** (2009) 1 (cit. on pp. 23, 57, 75).

[41] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak and Z. G. Ives, "DBpedia: A Nucleus for a Web of Open Data", *Proceedings of the 6th International Semantic Web Conference ISWC + ASWC, Busan, Korea, November 11-15*, 2007 722 (cit. on pp. 23, 53).

[42] D. Vrandecic and M. Krötzsch, *Wikidata: a free collaborative knowledgebase*, Commun. ACM **57** (2014) 78 (cit. on p. 23).

[43] H. Paulheim, *Knowledge graph refinement: A survey of approaches and evaluation methods*, Semantic Web **8** (2017) 489 (cit. on pp. 23, 53, 58).

[44] L. Ehrlinger and W. Wöß, "Towards a Definition of Knowledge Graphs", *Posters and Demos Track of the 12th International Conference on Semantic Systems SEMANTiCS, Leipzig, Germany, September 12-15*, 2016 (cit. on p. 24).

[45] C. Gutierrez, C. A. Hurtado, A. O. Mendelzon and J. Pérez, *Foundations of Semantic Web databases*, Journal of Computer and System Sciences **77** (2011) 520 (cit. on p. 24).

[46] M. Lenzerini, "Data Integration: A Theoretical Perspective", *Proceedings of the 21st ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 3-5, Madison, Wisconsin, USA*, 2002 233 (cit. on pp. 24, 138).

[47] N. Ide and J. Pustejovsky, "What does interoperability mean, anyway? Toward an operational definition of interoperability for language technology", *Proceedings of the 2nd International Conference on Global Interoperability for Language Resources. Hong Kong, China*, 2010 (cit. on p. 24).

[48] M. Bergman, *Sources and classification of semantic heterogeneities*, Web Blog: AI3-Adaptive Information, Adaptive Innovation, Adaptive Infrastructure (2006) (cit. on p. 24).

[49] M. of Industry, I. technology of China (MIIT) and S. A. of China (SAC), *National Intelligent Manufacturing Standard System Construction Guidelines*, tech. rep., Standardization Administration of China (SAC), 2015 (cit. on pp. 25, 74).

[50] M. Schleipen, M. Damm, R. Henßen, A. Lüder, N. Schmidt, O. Sauer and S. Hoppe, "OPC UA and AutomationML–collaboration partners for one common goal: Industry 4.0", *Proceedings of the 3rd AutomationML User Conference, Blomberg, Germany*, 2014 (cit. on p. 26).

[51] S. Biffl, O. Kovalenko, A. Lüder, N. Schmidt and R. Rosendahl, "Semantic mapping support in AutomationML", *Proceedings of the 19th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, Barcelona, Spain, September 16-19*, IEEE, 2014 1, ISBN: 978-1-4799-4846-8 (cit. on p. 26).

[52] O. Kovalenko and J. Euzenat, "Semantic Matching of Engineering Data Structures", *Semantic Web for Intelligent Engineering Applications*, Springer, 2016 (cit. on p. 26).

[53] S. P. Gardner, *Ontologies and semantic data integration*, Drug discovery today **10** (2005) 1001 (cit. on p. 27).

[54] G. Gottlob, G. Orsi, A. Pieris and M. Simkus, "Datalog and Its Extensions for Semantic Web Databases", *Proceedings of the 8th International Summer School. Reasoning Web. Semantic Technologies for Advanced Query Answering, Vienna, Austria, September 3-8*, 2012 54 (cit. on p. 27).

[55] S. H. Bach, M. Broecheler, B. Huang and L. Getoor, *Hinge-Loss Markov Random Fields and Probabilistic Soft Logic*, **arXiv:1505.04406 [cs.LG]** (2015) (cit. on p. 28).

[56] A. Kimmig, S. Bach, M. Broecheler, B. Huang and L. Getoor, "A short introduction to Probabilistic Soft Logic", *Proceedings of the NIPS Workshop on Probabilistic Programming: Foundations and Applications*, 2012 1 (cit. on p. 28).

[57] A. Memory, A. Kimmig, S. H. Bach, L. Raschid and L. Getoor, "Graph Summarization in Annotated Data Using Probabilistic Soft Logic", *Proceedings of the 8th International Workshop on Uncertainty Reasoning for the Semantic Web, Boston, USA, November 11*, 2012 75 (cit. on p. 29).

[58] F. Scharffe and A. Zimmermann, *D2. 2.10: Expressive alignment language and implementation*, Deliverable D2 **2** (2007) (cit. on p. 29).

[59] F. Scharffe, O. Zamazal and D. Fensel, *Ontology alignment design patterns*, Knowledge and Information Systems **40** (2014) 1, ISSN: 0219-3116 (cit. on p. 29).

[60] J. David, J. Euzenat, F. Scharffe and C. T. dos Santos, *The Alignment API 4.0*, Semantic Web **2** (2011) 3, URL: `https://doi.org/10.3233/SW-2011-0028` (cit. on p. 29).

[61] D. Collarana, M. Galkin, I. T. Ribón, M. Vidal, C. Lange and S. Auer, "MINTE: semantically integrating RDF graphs", *Proceedings of the 7th International Conference on Web Intelligence, Mining and Semantics, WIMS, Amantea, Italy, June 19-22*, 2017 22:1 (cit. on pp. 29, 32, 51).

[62] G. De Giacomo, D. Lembo, M. Lenzerini, A. Poggi and R. Rosati, "Using Ontologies for Semantic Data Integration", *A Comprehensive Guide Through the Italian Database Research Over the Last 25 Years*, 2018 187 (cit. on p. 31).

[63] M. Cheatham and C. Pesquita, "Semantic Data Integration", *Handbook of Big Data Technologies*, 2017 263 (cit. on p. 31).

[64] N. F. Noy, *Semantic Integration: A Survey Of Ontology-Based Approaches*, SIGMOD Record **33** (2004) 65 (cit. on p. 31).

[65] I. F. Cruz and H. Xiao, *The role of ontologies in data integration*, Engineering intelligent systems for electrical engineering and communications **13** (2005) 245 (cit. on p. 31).

[66] C. A. Knoblock, P. Szekely, J. L. Ambite, S. Gupta, A. Goel, M. Muslea, K. Lerman, M. Taheriyan and P. Mallick, "Semi-Automatically Mapping Structured Sources into the Semantic Web", *Proceedings of the 9th Extended Semantic Web Conference, ESWC, Heraklion, Crete, Greece, May 27-31*, 2012 (cit. on pp. 31, 40).

[67] J. Volz, C. Bizer, M. Gaedke and G. Kobilarov, "Silk - A Link Discovery Framework for the Web of Data", *Proceedings of the Workshop on Linked Data on the Web LDOW Madrid, Spain, April 20*, vol. 538, CEUR Workshop Proceedings, CEUR-WS.org, 2009 (cit. on pp. 32, 40).

[68] P. N. Mendes, H. Muhleisen and C. Bizer, "Sieve: linked data quality assessment and fusion", *Proceedings of the Joint EDBT/ICDT Workshops, Berlin, Germany, March 30*, 2012 116 (cit. on p. 32).

[69] E. Rahm, "The Case for Holistic Data Integration", *Proceedings of the 20th East European Conference, Advances in Databases and Information Systems ADBIS, Prague, Czech Republic, August 28-31*, 2016 11 (cit. on p. 32).

[70] A. Schultz, A. Matteini, R. Isele, C. Bizer and C. Becker, "LDIF - Linked Data Integration Framework", *Proceedings of the 2nd International Workshop on Consuming Linked Data (COLD2011), Bonn, Germany, October 23*, 2011 (cit. on p. 32).

[71] N. Chungoora, A. Cutting-Decelle, R. Young, G. Gunendran, Z. Usman, J. Harding and K. Case, *Towards the ontology-based consolidation of production-centric standards*, International Journal of Production Research **51** (2013) 327 (cit. on p. 32).

[72] J. Hodges, K. Garcia and S. Ray, *Semantic Development and Integration of Standards for Adoption and Interoperability*, IEEE Computer **50** (2017) 26 (cit. on pp. 32, 34, 67).

[73] A. J. C. Trappey, C. V. Trappey, U. H. Govindarajan, J. J. Sun and A. C. Chuang, *A Review of Technology Standards and Patent Portfolios for Enabling Cyber-Physical Systems in Advanced Manufacturing*, IEEE Access **4** (2016) 7356 (cit. on p. 33).

[74] F. Ansari, M. Khobreh, U. Seidenberg and W. Sihn, *A problem-solving ontology for human-centered Cyber-Physical production systems*, CIRP Journal of Manufacturing Science and Technology **22** (2018) 91 (cit. on p. 33).

[75] C. Legat, C. Seitz, S. Lamparter and S. Feldmann, “Semantics to the shop floor: towards ontology modularization and reuse in the automation domain”, vol. 47, 3, Elsevier, 2014 3444 (cit. on p. 33).

[76] Y. Lu, K. C. Morris and S. Frechette, “Standards landscape and directions for smart manufacturing systems”, *Proceedings of the 11th IEEE International Conference on Automation Science and Engineering, CASE, Gothenburg, Sweden, August 24-28*, 2015 998 (cit. on pp. 33, 34).

[77] E. Darmois, O. Elloumi, P. Guillemin and P. Moretto, *IoT Standards–State-of-the-Art Analysis*, Digitising the Industry Internet of Things Connecting the Physical, Digital and Virtual Worlds (2012) 978 (cit. on p. 34).

[78] R. Herzog, M. Jacoby and I. P. Zarko, *Semantic interoperability in IoT-based automation infrastructures*, Automatisierungstechnik **64** (2016) 742 (cit. on p. 34).

[79] N. Siltala, *Formal Digital Description of Production Equipment Modules for supporting System Design and Deployment*, PhD thesis: Tampere University of Technology, 2016 (cit. on p. 34).

[80] Q. Li, H. Jiang, Q. Tang, Y. Chen, J. Li and J. Zhou, “Smart Manufacturing Standardization: Reference Model and Standards Framework”, *On the Move to Meaningful Internet Systems: OTM Workshops*, Springer, 2016 16 (cit. on pp. 34, 74).

[81] Y. Zhao, Q. Liu and W. Xu, “Open Industrial Knowledge Graph Development for Intelligent Manufacturing Service Matchmaking”, *Proceedings of the International Conference on Industrial Informatics-Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII)*, IEEE, 2017 194 (cit. on p. 34).

[82] C. Galinski, *Interoperability of Metadata. Semantic Interoperability*, tech. rep., International Information Centre for Terminology, Austria, 2005 (cit. on p. 34).

[83] G. Engel, T. Greiner and S. Seifert, *Ontology-Assisted Engineering of Cyber-Physical Production Systems in the Field of Process Technology*, IEEE Transactions on Industrial Informatics **14** (2018) 2792 (cit. on p. 34).

[84] H. Panetto, M. Dassisti and A. Tursi, *ONTO-PDM: Product-driven ONTOlogy for Product Data Management interoperability within manufacturing process environment*, Advanced Engineering Informatics **26** (2012) 334 (cit. on pp. 35, 41).

[85] M. Dassisti, H. Panetto, A. Tursi and M. De Nicolo, “Ontology-based model for production-control systems interoperability”, *The 5th CIRP Digital Enterprise Technology Conference*, 2008 527 (cit. on p. 35).

[86] M. Sabou, O. Kovalenko and P. Novak, “Semantic Modelling and Acquisition of Engineering Knowledge”, *Semantic Web Technologies for Intelligent Engineering Applications*, Springer, 2016 105 (cit. on pp. 35, 67, 70, 93).

[87] V. Damjanovic, W. Behrendt, M. Plossnig and M. Holzapfel, "Developing Ontologies for Collaborative Engineering in Mechatronics", *Proceedings of the 4th European Semantic Web Conference, ESWC. The Semantic Web: Research and Applications, Innsbruck, Austria, June 3-7*, 2007 190 (cit. on pp. 35, 37).

[88] A. L. Szejka and O. C. Junior, *The Application of Reference Ontologies for Semantic Interoperability in an Integrated Product Development Process in Smart Factories*, Procedia Manufacturing **11** (2017) 1375 (cit. on p. 35).

[89] R. M., S. R. and T. M., *EDDL – Electronic Device Description Language*, München, Oldenburg Industrieverlag (2002) (cit. on p. 36).

[90] J. Haupert, M. Seissler, B. Kiesel, B. Schennerlein, S. Horn, D. Schreiber and R. Barthel, *Object Memory Modeling*, Incubator Group Report, World Wide Web Consortium (W3C), 2011, URL: https://www.w3.org/2005/Incubator/omm/XGR-omm-20111026/ (cit. on p. 36).

[91] T. Aruvali, W. Maass and T. Otto, *Digital Object Memory Based Monitoring Solutions in Manufacturing Processes*, Procedia Engineering **69** (2014) 449 (cit. on p. 36).

[92] D. Brock, T. Milne, Y. Kang and B. Lewis, *The physical markup language*, Auto-ID Center White Paper MIT-AUTOID-WH-003 (2001) (cit. on p. 36).

[93] S. Janzen and W. Maass, "Smart product description object (SPDO)", *Poster Proceedings of the 5th International Conference on Formal Ontology in Information Systems (FOIS), Saarbrücken, Germany, October 31st - November 3rd*, 2008 (cit. on pp. 36, 73).

[94] E. Tantik and A. Reiner, *Integrated data model and structure for the asset administration shell in Industrie 4.0*, Procedia CIRP **60** (2017) 86 (cit. on p. 35).

[95] F. Pethig, O. Niggemann and A. Walter, "Towards Industrie 4.0 compliant configuration of condition monitoring services", *Proceedings of the 15th IEEE International Conference on Industrial Informatics, INDIN, Emden, Germany, July 24-26*, 2017 271 (cit. on pp. 35, 73).

[96] C. Diedrich, A. Belyaev, T. Schroder, J. Vialkowitsch, A. Willmann, T. Usländer, H. Koziolek, J. Wende, F. Pethig and O. Niggemann, "Semantic interoperability for asset communication within smart factories", *Proceedings of the 22nd IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, Limassol, Cyprus, September 12-15*, 2017 1 (cit. on pp. 35, 73).

[97] S. Runde, G. Wolf and M. Braun, "EDDL and semantic web - From field device integration (FDI) to Future Device Management (FDM)", *ETFA*, ed. by C. Seatzu, IEEE, 2013 1, ISBN: 978-1-4799-0864-6 (cit. on p. 36).

[98] J. Haupert, "DOMeMan: a framework for representation, management, and utilization of digital object memories", *Proceedings of the 9th International Conference on Intelligent Environments (IE), Greece, July 16-17*, IEEE, 2013 84 (cit. on p. 36).

[99] J. Haupert and M. Schneider, "The Object Memory Server for Semantic Product Memories", *SemProM - Foundations of Semantic Product Memories for the Internet of Things*, Springer, 2013 175 (cit. on p. 36).

[100] S. Bergweiler, *Intelligent Manufacturing based on Self-Monitoring Cyber-Physical Systems*, UBICOMM 2015 (2015) 121 (cit. on p. 36).

[101] L. Abele, C. Legat, S. Grimm and A. W. Mueller, "Ontology-based validation of plant models", *Proceedings of the 11th IEEE International Conference on Industrial Informatics, INDIN, Bochum, Germany, July 29-31*, 2013 236, ISBN: 978-1-4799-0752-6 (cit. on pp. 37, 86).

[102] A. Björkelund, J. Malec, K. Nilsson and P. Nugues, "Knowledge and skill representations for robotized production", *Proceedings of the 18th IFAC Congress, Milan*, 2011 (cit. on pp. 37, 86).

[103] J. Persson, A. Gallois, A. Björkelund, L. Hafdell, M. Haage, J. Malec, K. Nilsson and P. Nugues, "A Knowledge Integration Framework for Robotics", *ISR/ROBOTIK 2010, Proceedings for the joint conference of ISR (41st Internationel Symposium on Robotics) und ROBOTIK (6th German Conference on Robotics), 7-9 June, Munich, Germany - Parallel to AUTOMATICA*, 2010 1 (cit. on pp. 37, 86).

[104] M. Glawe, C. Tebbe, A. Fay and K. Niemann, "Knowledge-based Engineering of Automation Systems using Ontologies and Engineering Data", *Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering, and Knowledge Management, IC3K, Volume 2: KEOD, Lisbon, Portugal, November 9-11*, 2015 291 (cit. on pp. 37, 86).

[105] Y. Hua, S. Zander, M. Bordignon and B. Hein, "From AutomationML to ROS: A model-driven approach for software engineering of industrial robotics using ontological reasoning", *Proceedings of the 21st IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, Berlin, Germany, September 6-9*, 2016 1 (cit. on pp. 37, 86).

[106] Y. Ye, D. Yang, Z. Jiang and L. Tong, *An ontology-based architecture for implementing semantic integration of supply chain management*, International Journal of Computer Integrated Manufacturing **21** (2008) 1 (cit. on p. 38).

[107] M. Fayez, L. Rabelo and M. Mollaghasemi, "Ontologies for supply chain simulation modeling", *Proceedings of the 37th Conference on Winter Simulation*, 2005 (cit. on p. 38).

[108] J. Leukel and S. Kirn, "A supply chain management approach to logistics ontologies in information systems", *Business Information Systems*, Springer, 2008 (cit. on p. 38).

[109] O. Sakka, P.-A. Millet and V. Botta-Genoulaz, *An ontological approach for strategic alignment: a supply chain operations reference case study*, International Journal of Computer Integrated Manufacturing **24** (2011), ISSN: 0951-192X (cit. on p. 38).

[110] M. Zdravković, H. Panetto, M. Trajanović and A. Aubry, *An approach for formalising the supply chain operations*, Enterprise Information Systems **5** (2011) (cit. on pp. 38, 39).

[111] Y. Lu, H. Panetto, Y. Ni and X. Gu, *Ontology alignment for networked enterprise information system interoperability in supply chain environment*, International Journal of Computer Integrated Manufacturing **26** (2013) 140 (cit. on p. 38).

[112] M. Zdravković, M. Trajanović, H. Panetto, A. Aubry and M. Lezoche, “Ontology-based supply chain process configuration”, *Proceedings of the 34th International Conference on Production Engineering, ICPE*, 2011 (cit. on p. 38).

[113] H. Panetto, M. Dassisti and A. Tursi, *ONTO-PDM: Product-driven ONTOlogy for Product Data Management interoperability within manufacturing process environment*, Advanced Engineering Informatics **26** (2012) (cit. on p. 38).

[114] T. Grubic and I. S. Fan, *Integrating process and ontology for supply chain modelling*, International Journal of Computer Integrated Manufacturing **24** (2011) 847, ISSN: 0951-192X (cit. on p. 39).

[115] N. Schmidt, A. Lüder, R. Rosendahl, D. Ryashentseva, M. Foehr and J. Vollmar, “Surveying integration approaches for relevance in Cyber-Physical Production Systems”, *Proceedings of the 20th IEEE Conference on Emerging Technologies and Factory Automation, ETFA, Luxembourg, September 8-11*, 2015 1 (cit. on p. 39).

[116] S. Biffl, A. Lüder and D. Winkler, “Multi-Disciplinary Engineering for Industrie 4.0: Semantic Challenges and Needs”, *Semantic Web Technologies for Intelligent Engineering Applications*, 2016 17 (cit. on p. 39).

[117] M. Schleipen and M. Okon, “The CAEX tool suite - User assistance for the use of standardized plant engineering data exchange”, *Proceedings of the 15th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, Bilbao, Spain, September 13-16*, 2010 (cit. on p. 39).

[118] F. Himmler, “Function Based Engineering with AutomationML - Towards better standardization and seamless process integration in plant engineering”, *12th International Tagung Wirtschaftsinformatik, WI*, 2015 (cit. on p. 39).

[119] L. Bi and Z. Jiao, “An Information Integration Framework Based on XML to Support Mechatronics Multi-disciplinary Design”, *IEEE Conference on Robotics, Automation and Mechatronics, RAM, 21-24 September, Chengdu, China*, 2008 175 (cit. on pp. 39, 110).

[120] A. Lüder, N. Schmidt, R. Rosendahl and M. John, “Integrating different information types within AutomationML”, *Proceedings of the IEEE Emerging Technology and Factory Automation, ETFA, Barcelona, Spain, September 16-19*, 2014 1 (cit. on pp. 39, 110).

[121] K. Chen, B. Jonathan, J. H. Panchal and D. Schaefer, “A framework for integrated design of mechatronic systems”, *Collaborative Design and Planning for Digital Manufacturing*, Springer, 2009 37 (cit. on p. 39).

[122] M. Prösser, P. R. Moore, X. Chen, C. Wong and U. Schmidt, *A new approach towards systems integration within the mechatronic engineering design process of manufacturing systems*, International Journal Computer Integrated Manufacturing **26** (2013) 806 (cit. on pp. 39, 109).

[123] P. Bihani and R. Drath, "Concept for AutomationML-based interoperability between multiple independent engineering tools without semantic harmonization: Experiences with AutomationML", *Proceedings of the 22nd IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, Limassol, Cyprus, September 12-15, 2017*, 2017 1 (cit. on p. 39).

[124] F. J. Ekaputra, M. Sabou, E. Serral, E. Kiesling and S. Biffl, *Ontology-Based Data Integration in Multi-Disciplinary Engineering Environments: A Review*, Open Journal of Information Systems **4** (2017) 1 (cit. on pp. 40, 41).

[125] L. Abele and S. Grimm, "Knowledge-based integration of industrial plant models", *Industrial Electronics Society, IECON 2013-39th Annual Conference of the IEEE*, IEEE, 2013 4392 (cit. on p. 41).

[126] C. Brecher, D. Ozdemir, J. Feng, W. Herfs, K. Fayzullin, M. H. M and A. AM Muller, *Integration of software tools with heterogeneous data structures in production plant lifecycles*, IFAC Proceedings Volumes **43** (2010) 48 (cit. on p. 41).

[127] M. Graube, L. Urbas and J. Hladik, "Integrating industrial middleware in Linked Data collaboration networks", *Proceedings of the 21st IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, Berlin, Germany, September 6-9*, 2016 1 (cit. on p. 41).

[128] C. Hennig, A. Viehl, B. Kampgen and H. Eisenmann, "Ontology-Based Design of Space Systems", *Proceedings of the 15th International Semantic Web Conference ISWC, Kobe, Japan, October 17-21*, 2016 308 (cit. on p. 41).

[129] 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 **33** (2007) 59 (cit. on pp. 40, 41).

[130] P. Novak and R. Sindelar, "Ontology-based industrial plant description supporting simulation model design and maintenance", *Industrial Electronics Society, IECON 2013-39th Annual Conference of the IEEE*, IEEE, 2013 6866 (cit. on p. 41).

[131] M. Sabou, F. Ekaputra, O. Kovalenko and S. Biffl, "Supporting the engineering of Cyber-Physical production systems with the AutomationML analyzer", *Proceedings of the 1st International Workshop on Cyber-Physical Production Systems*, IEEE, 2016 1 (cit. on pp. 41, 111).

[132] S. Softic, M. Rosenberger, A. Denger, J. Fritz and A. Stocker, "Semantically based visual tracking of engineering tasks in automotive product lifecycle", *Proceedings of the 13th International Conference on Knowledge Management and Knowledge Technologies*, ACM, 2013 36 (cit. on p. 41).

[133] W. Terkaj and M. Urgo, "Ontology-based modeling of production systems for design and performance evaluation", *Proceedings of the 12th International Conference on Industrial Informatics, INDIN, Porto Alegre, RS, Brazil, July 27-30*, 2014 748 (cit. on p. 41).

[134] J. Angele and M. Gesmann, "Data Integration using Semantic Technology: A use case", *Proceedings of the 2nd International Conference of Rules and Rule Markup Languages for the Semantic Web (RuleML), Athens, Georgia, USA, November 10-11*, 2006 58 (cit. on pp. 40–42).

[135] S. Feldmann, S. J. Herzig, K. Kernschmidt, T. Wolfenstetter, D. Kammerl, A. Qamar, U. Lindemann, H. Krcmar, C. Paredis and B. Vogel-Heuser, *Towards effective management of inconsistencies in model-based engineering of automated production systems*, IFAC-PapersOnLine **48** (2015) 916 (cit. on p. 41).

[136] O. Kovalenko, E. Serral, M. Sabou, F. J. Ekaputra, D. Winkler and S. Biffl, "Automating Cross-Disciplinary Defect Detection in Multi-disciplinary Engineering Environments", *Proceedings of the 19th International Conference of Knowledge Engineering and Knowledge Management, EKAW, Linkoping, Sweden, November 24-28*, 2014 238 (cit. on p. 41).

[137] E. Kharlamov, N. Solomakhina, O. L. Ozccep, D. Zheleznyakov, T. Hubauer, S. Lamparter, M. Roshchin, A. Soylu and S. Watson, "How Semantic Technologies Can Enhance Data Access at Siemens Energy", *Proceedings of the 13th International Semantic Web Conference, ISWC, Riva del Garda, Italy, October 19-23*, 2014 601 (cit. on p. 41).

[138] P. Aarnio, I. Seilonen and M. Friman, "Semantic repository for case-based reasoning in CBM services", *Proceedings of the 19th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, Barcelona, Spain, September 16-19*, 2014 1 (cit. on p. 41).

[139] H. Dibowski and K. Kabitzsch, "Ontology-based Device Descriptions and triple store based device repository for automation devices", *Proceedings of 15th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, Bilbao, Spain, September 13-16*, 2010 1 (cit. on p. 41).

[140] V. Dubinin, V. Vyatkin, C. Yang and C. Pang, "Automatic generation of automation applications based on ontology transformations", *Proceedings of the 19th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, Barcelona, Spain, September 16-19*, 2014 1 (cit. on p. 41).

[141] F. J. Ekaputra, "Knowledge Change Management and Analysis in Engineering", *Semantic Web Technologies for Intelligent Engineering Applications*, 2016 159 (cit. on p. 41).

[142] M. Imran and R. Young, *Reference ontologies for interoperability across multiple assembly systems*, International Journal of Production Research **54** (2016) 5381 (cit. on pp. 40, 41).

[143] H. Lin and J. A. Harding, *A manufacturing system engineering ontology model on the semantic web for inter-enterprise collaboration*, Computers in Industry **58** (2007) 428 (cit. on p. 41).

[144] J. Persson, A. Gallois, A. Björkelund, L. Hafdell, M. Haage, J. Malec, K. Nilsson and P. Nugues, “A Knowledge Integration Framework for Robotics”, *41st International Symposium on Robotics and ROBOTIK 2010*, 2010 (cit. on p. 41).

[145] 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 **8** (2017) 471 (cit. on pp. 40, 139).

[146] C. Bizer and A. Seaborne, “D2RQ - Treating Non-RDF Databases as Virtual RDF Graphs”, *ISWC2004 (posters)*, 2004 (cit. on pp. 40, 81).

[147] O. Kovalenko and J. Euzenat, “Semantic Matching of Engineering Data Structures”, *Semantic Web Technologies for Intelligent Engineering Applications*, 2016 137 (cit. on p. 40).

[148] T. Adams, J. Dullea, P. Clark, S. Sripada and T. Barrett, “Semantic integration of heterogeneous information sources using a knowledge-based system”, *Proceedings of the 5th International Conference on Computer Science and Informatics (CS&I)*, Citeseer, 2000 (cit. on p. 42).

[149] E. Kharlamov, T. Mailis, G. Mehdi, C. Neuenstadt, Ö. Özçep, M. Roshchin, N. Solomakhina, A. Soylu, C. Svingos, S. Brandt et al., *Semantic access to streaming and static data at Siemens*, Web Semantics (2017, in press) (cit. on p. 42).

[150] E. Kharlamov, S. Brandt, E. Jimenez-Ruiz, Y. Kotidis, S. Lamparter, T. Mailis, C. Neuenstadt, Ö. Özçep, C. Pinkel and C. Svingos, “Ontology-based integration of streaming and static relational data with OPTIQUE”, *SIGMOD*, 2016 (cit. on p. 42).

[151] A. Fishkin, *Industrial Knowledge Graph at Siemens*, tech. rep., CERN Openlab Technical Workshop, 2018 (cit. on p. 42).

[152] 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 **30** (2017) 1043 (cit. on p. 42).

[153] S. Schabus and J. Scholz, “Semantically Annotated Manufacturing Data to support Decision Making in Industry 4.0: A Use-Case Driven Approach”, *Data Science–Analytics and Applications*, Springer, 2017 97 (cit. on p. 42).

[154] E. Kharlamov, D. Hovland, E. Jiménez-Ruiz, D. Lanti, H. Lie, C. Pinkel, M. Rezk, M. G. Skjæveland, E. Thorstensen, G. Xiao et al., “Ontology based access to exploration data at Statoil”, *ISWC*, 2015 (cit. on p. 43).

[155] N. Rychtyckyj, V. Raman, B. Sankaranarayanan, P. S. Kumar and D. Khemani, “Ontology re-engineering: a case study from the automotive industry”, *AAAI*, 2016 (cit. on p. 43).

[156] D. Ostrowski, N. Rychtyckyj, P. MacNeille and M. Kim, “Integration of big data using semantic web technologies”, *ICSC*, 2016 (cit. on p. 43).

[157] M. Kim, S.-T. Wang, D. Ostrowski, N. Rychtyckyj and P. Macneille, *Technology Outlook: Federated Ontologies and Industrial Applications*, Semantic Computing **10** (2016) (cit. on p. 43).

[158] W. Greenly, C. Sandeman-Craik, Y. Otero and J. Streit, *Case Study: Contextual Search for Volkswagen and the Automotive Industry*, 2011, URL: https://www.w3.org/2001/sw/sweo/public/UseCases/Volkswagen/ (cit. on p. 43).

[159] J. Golebiowska, R. Dieng-Kuntz, O. Corby and D. Mousseau, "Building and exploiting ontologies for an automobile project memory", *Proceedings of the 1st International Conference on Knowledge Capture, K-CAP, October 21-23, Victoria, BC, Canada*, 2001 (cit. on p. 43).

[160] M. Rodriguez-Muro, R. Kontchakov and M. Zakharyaschev, "Ontology-based data access: Ontop of databases", *Proceedings of 12th International Semantic Web Conference ISWC, Sydney, NSW, Australia, October 21-25*, 2013 (cit. on p. 43).

[161] B. Vogel-Heuser and D. Hess, *Guest editorial Industry 4.0–prerequisites and visions*, IEEE Transactions on Automation Science and Engineering **13** (2016) 411 (cit. on p. 43).

[162] I. Grangel-González, L. Halilaj, M. Vidal, S. Lohmann and S. Auer, "Towards a Knowledge Graph of Industry 4.0 Standards", 2018 (cit. on p. 45).

[163] I. Grangel-González, P. Baptista, L. Halilaj, S. Lohmann, M. Vidal, C. Mader and S. Auer, "The industry 4.0 standards landscape from a semantic integration perspective", *Proceedings of the 22nd IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, Limassol, Cyprus, September 12-15*, 2017 1 (cit. on p. 45).

[164] K. M. Endris, M. Galkin, I. Lytra, M. N. Mami, M. Vidal and S. Auer, "MULDER: Querying the Linked Data Web by Bridging RDF Molecule Templates", *Proceedings of the 28th International Conference of Database and Expert Systems Applications DEXA, Lyon, France, August 28-31*, 2017 3 (cit. on p. 52).

[165] S. Kozar and P. Kadera, "Integration of IEC 61499 with OPC UA", *Proceedings of the 21st IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, Berlin, Germany, September 6-9*, 2016 1 (cit. on p. 54).

[166] C. Pang and V. Vyatkin, "IEC 61499 function block implementation of Intelligent Mechatronic Component", *Proceedings of the 8th IEEE International Conference on Industrial Informatics, INDIN, Osaka, Japan, July 13-16*, 2010 1124 (cit. on pp. 56, 57).

[167] G. Töpper, M. Knuth and H. Sack, "DBpedia ontology enrichment for inconsistency detection", *Proceedings of the 8th International Conference on Semantic Systems I-SEMANTICS, Graz, Austria, September 5-7*, 2012 33 (cit. on p. 61).

[168] E. Kharlamov, B. C. Grau, E. Jimenez-Ruiz, S. Lamparter, G. Mehdi, M. Ringsquandl, Y. Nenov, S. Grimm, M. Roshchin and I. Horrocks, "Capturing Industrial Information Models with Ontologies and Constraints", *Proceedings of the 15th International Semantic Web Conference ISWC, Kobe, Japan, October 17–21*, Springer International Publishing, 2016 325 (cit. on pp. 67, 131).

[169] M. Sabou, O. Kovalenko, F. J. Ekaputra and S. Biffl, "Semantic Web Solutions in Engineering", *Semantic Web Technologies for Intelligent Engineering Applications*, ed. by S. Biffl and M. Sabou, Springer International Publishing, 2016 281, ISBN: 978-3-319-41490-4 (cit. on p. 67).

[170] M. Sabou, "An Introduction to Semantic Web Technologies", *Semantic Web Technologies for Intelligent Engineering Applications*, ed. by S. Biffl and M. Sabou, Springer International Publishing, 2016 53, ISBN: 978-3-319-41490-4 (cit. on p. 67).

[171] O. Kovalenko, I. Grangel-González, M. Sabou, A. Lüder, S. Biffl, S. Auer and M.-E. Vidal, "AutomationML Ontology: Modeling Cyber-Physical Systems for Industry 4.0", 2018 (cit. on pp. 68, 86).

[172] N. Petersen, I. Grangel-González, G. Coskun, S. Auer, M. Frommhold, S. Tramp, M. Lefrançois and A. Zimmermann, "SCORVoc: Vocabulary-Based Information Integration and Exchange in Supply Networks", *Proceedings of the Tenth IEEE International Conference on Semantic Computing, ICSC, Laguna Hills, CA, USA, February 4-6*, 2016 132 (cit. on pp. 68, 97).

[173] I. Grangel-González, L. Halilaj, G. Coskun, S. Auer, D. Collarana and M. Hoffmeister, "Towards a Semantic Administrative Shell for Industry 4.0 Components", *Proceedings of the Tenth IEEE International Conference on Semantic Computing, ICSC, Laguna Hills, CA, USA, February 4-6*, 2016 230 (cit. on p. 68).

[174] I. Grangel-González, L. Halilaj, G. Coskun and S. Auer, "Towards Vocabulary Development by Convention", *Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering, and Knowledge Management, IC3K, Volume 2: KEOD, Lisbon, Portugal, November 12-14*, SCITEPRESS, 2015 334 (cit. on pp. 68, 69).

[175] I. Grangel-González, L. Halilaj, S. Auer, S. Lohmann, C. Lange and D. Collarana, "An RDF-based approach for implementing industry 4.0 components with Administration Shells", *Proceedings of the 21st IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, Berlin, Germany, September 6-9*, 2016 (cit. on p. 68).

[176] M. Uschold and M. Gruninger, *Ontologies: Principles, methods and applications*, Knowledge engineering review **11** (1996) 93 (cit. on pp. 68, 81, 95, 103).

[177] L. Halilaj, N. Petersen, I. Grangel-González, C. Lange, S. Auer, G. Coskun and S. Lohmann, "VoCol: An Integrated Environment to Support Version-Controlled Vocabulary Development", *Proceedings of the 20th International Conference of Knowledge Engineering and Knowledge Management EKAW, Bologna, Italy, November 19-23*, 2016 303 (cit. on p. 68).

[178] C. Pedrinaci, J. S. Cardoso and T. Leidig, "Linked USDL: A Vocabulary for Web-Scale Service Trading", *Proceedings of the 11th International Conference, The Semantic Web: Trends and Challenges, ESWC, Anissaras, Crete, Greece, May 25-29*, Springer, 2014 68 (cit. on p. 69).

[179] M. Poveda-Villalon, "A reuse-based lightweight method for developing linked data ontologies and vocabularies", *The Semantic Web: Research and Applications*, Springer, 2012 833 (cit. on p. 69).

[180] T. Heath and C. Bizer, *Linked data: Evolving the web into a global data space*, Synthesis lectures on the semantic web: theory and technology **1** (2011) 1 (cit. on p. 69).

[181] A. Gangemi and V. Presutti, "Ontology Design Patterns", *Handbook on Ontologies*, 2009 221 (cit. on p. 69).

[182] D. Schober, B. Smith, S. E. Lewis, W. Kusnierczyk, J. Lomax, C. Mungall, C. F. Taylor, P. Rocca-Serra and S.-A. Sansone, *Survey-based naming conventions for use in OBO Foundry ontology development*, BMC bioinformatics **10** (2009) 125 (cit. on pp. 70, 72).

[183] H. Rijgersberg, M. van Assem and J. L. Top, *Ontology of units of measure and related concepts*, Semantic Web **4** (2013) 3 (cit. on pp. 70, 90).

[184] S. Peroni, D. Shotton and F. Vitali, *Tools for the automatic generation of ontology documentation: a task-based evaluation*, International Journal on Semantic Web and Information Systems **9** (2013) 21 (cit. on p. 71).

[185] D. Schober, I. Tudose, V. Svatek and M. Boeker, *OntoCheck: verifying ontology naming conventions and metadata completeness in Protégé 4.*, Journal Biomedical Semantics **3** (2012) S4 (cit. on p. 72).

[186] V. Svátek and O. Šváb-Zamazal, "Entity naming in semantic web ontologies: Design patterns and empirical observations", Znalosti, 2010 (cit. on p. 72).

[187] E. Montiel-Ponsoda, D. V. Suero, B. Villazon-Terrazas, G. Dunsire, E. E. Rodriguez and A. Gomez-Perez, *Style guidelines for naming and labeling ontologies in the multilingual web*, (2011) (cit. on p. 72).

[188] V. Svatek, O. Svab-Zamazal and V. Presutti, "Ontology naming pattern sauce for (human and computer) gourmets", *Workshop on Ontology Patterns*, 2009 171 (cit. on p. 72).

[189] J. Gracia, E. Montiel-Ponsoda, P. Cimiano, A. Gomez-Perez, P. Buitelaar and J. McCrae, *Challenges for the multilingual web of data*, Journal of Web Semantics **11** (2012) 63 (cit. on p. 73).

[190] P. I. 4.0, *Architecture Alignment and Interoperability.*, tech. rep., Plattform Industrie 4.0, 2018, URL: `https://www.plattform-i40.de/I40/Redaktion/EN/Downloads/Publikation/rami40-an-introduction.pdf?__blob=publicationFile&v=7` (cit. on p. 74).

[191] K. Främling, M. Harrison, J. Brusey and J. Petrow, *Requirements on unique identifiers for managing product lifecycle information: comparison of alternative approaches*, International Journal of Computer Integrated Manufacturing **20** (2007) 715 (cit. on p. 74).

[192] O. Vermesan, P. Friess, P. Guillemin, S. Gusmeroli, H. Sundmaeker, A. Bassi, I. S. Jubert, M. Mazura, M. Harrison and M. Eisenhauer, *Internet of things strategic research roadmap*, Internet of Things: Global Technological and Societal Trends **1** (2011) 9 (cit. on p. 74).

[193] M. Abu-Elkheir, M. Hayajneh and N. A. Ali, *Data management for the internet of things: Design primitives and solution*, Sensors **13** (2013) 15582 (cit. on p. 74).

[194] R. Drath, *Datenaustausch in der Anlagenplanung mit AutomationML: Integration von CAEX, PLCopen XML und COLLADA*, Springer-Verlag, 2010 (cit. on p. 74).

[195] IEC, *IEC 61918:2013 Industrial communication networks - Installation of communication networks in industrial premises*, 2013, URL: https://webstore.iec.ch/publication/6100 (cit. on p. 74).

[196] W. Mahnke, S.-H. Leitner and M. Damm, *OPC unified architecture*, Springer Science & Business Media, 2009, ISBN: 978-3-540-68898-3 (cit. on p. 74).

[197] U. Enste and W. Mahnke, *OPC Unified Architecture*, Automatisierungstechnik **59** (2011) 397 (cit. on p. 74).

[198] M. Hillier, *The role of cultural context in multilingual website usability*, Electronic Commerce Research and Applications **2** (2003) 2 (cit. on p. 74).

[199] M. Thoma, T. Braun, C. Magerkurth and A. Antonescu, "Managing things and services with semantics: A survey", *Network Operations and Management Symposium (NOMS)*, IEEE, 2014 1 (cit. on p. 75).

[200] W. Wahlster, "Semantic technologies for mass customization", *Towards the Internet of Services: The THESEUS Research Program*, Springer, 2014 3 (cit. on p. 75).

[201] D. Bandyopadhyay and J. Sen, *Internet of things: Applications and challenges in technology and standardization*, Wireless Personal Communications **58** (2011) 49 (cit. on p. 75).

[202] B. Quilitz and U. Leser, "Querying Distributed RDF Data Sources with SPARQL", *ESWC*, ed. by S. Bechhofer, M. Hauswirth, J. Hoffmann and M. Koubarakis, vol. 5021, Lecture Notes in Computer Science, Springer, 2008 524, ISBN: 978-3-540-68233-2 (cit. on p. 76).

[203] ZVEI, Z. Elektrotechnik and Elektronikindustrie, *Examples of the Asset Administration Shell for Industrie 4.0 Components – Basic Part*, tech. rep., German Electrical and Electronic Manufacturers Association, 2017, URL: https://www.zvei.org/fileadmin/user_upload/Presse_und_Medien/Publikationen/2017/April/Asset_Administration_Shell/ZVEI_WP_Verwaltungschale_Englisch_Download_03.04.17.pdf (cit. on pp. 77, 84).

[204] P. Adolphs, S. Auer, H. Bedenbender, M. Billmann, M. Hankel, R. Heidel, M. Hoffmeister, H. Huhle, M. Jochem, M. Kiele, G. Koschnick, H. Koziolek, L. Linke, R. Pichler, F. Schewe, K. Schneider and B. Waser, *Structure of the Administration Shell. Continuation of the Development of the Reference Model for the Industrie 4.0 Component*, Status Report, ZVEI and VDI, 2016, URL: http://www.plattform-i40.de/I40/Redaktion/EN/Downloads/Publikation/structure-of-the-administration-shell.html (cit. on p. 78).

[205] A. Sirbu, M. Becker, S. Caminiti, B. D. Baets, B. Elen, L. Francis, P. Gravino, A. Hotho, S. Ingarra, V. Loreto, A. Molino, J. Mueller and J. Peters, *Participatory Patterns in an International Air Quality Monitoring Initiative*, PLOS ONE **10** (2015), ISSN: 1932-6203 (cit. on p. 81).

[206] ZVEI, Z. Elektrotechnik and Elektronikindustrie, *Interaction Model for Industrie 4.0 Components*, tech. rep., Federal Ministry for Economic Affairs and Energy (BMWi), 2016, URL: https://www.plattform-i40.de/I40/Redaktion/EN/Downloads/Publikation/interaction-model-I40-components.pdf?__blob=publicationFile&v=9 (cit. on pp. 83, 84).

[207] eClass e.V., *eClass Standardized Material and Service Classification*, 2016, URL: http://www.eclass.eu/ (cit. on pp. 85, 92).

[208] M. Hepp, "GoodRelations: An Ontology for Describing Products and Services Offers on the Web", *Proceedings of the 16th International Conference of Knowledge Engineering and Knowledge Management EKAW, Acitrezza, Italy, September 29 - October 23*, 2008 329 (cit. on p. 85).

[209] O. Kovalenko, M. Wimmer, M. Sabou, A. Lüder, F. J. Ekaputra and S. Biffl, "Modeling AutomationML: Semantic Web technologies vs. Model-Driven Engineering", *Proceedings of the 20th IEEE Conference on Emerging Technologies and Factory Automation, ETFA, Luxembourg, September 8-11*, 2015 1 (cit. on pp. 86, 89).

[210] O. Kovalenko, M. Wimmer, M. Sabou, A. Lüder, F. J. Ekaputra and S. Biffl, "Modeling AutomationML: Semantic Web technologies vs. Model-Driven Engineering", *Proceedings of the 20th IEEE Conference on Emerging Technologies and Factory Automation, ETFA, Luxembourg, September 8-11*, 2015 1 (cit. on p. 86).

[211] I. Grangel-González, D. Collarana, L. Halilaj, S. Lohmann, C. Lange, M.-E. Vidal and S. Auer, "Alligator: A Deductive Approach for the Integration of Industry 4.0 Standards", *Proceedings of the 20th International Conference of Knowledge Engineering and Knowledge Management, EKAW, Bologna, Italy, November 19-23*, 2016 (cit. on pp. 86, 89, 110).

[212] A. Gray and M. Sabou, *ISWC2016 Resources Track Review Instructions*, ISWC2016 (2016) (cit. on p. 86).

[213] V. Presutti, E. Blomqvist, E. Daga and A. Gangemi, "Pattern-Based Ontology Design", *Ontology Engineering in a Networked World*, ed. by M. C. Suarez-Figueroa, A. Gomez-Perez, E. Motta and A. Gangemi, Springer Berlin Heidelberg, 2012 35, ISBN: 978-3-642-24794-1 (cit. on p. 89).

[214] H. S. Pinto and J. P. Martins, *Ontologies: How can They be Built?*, Knowledge and Information Systems **6** (2004) 441, ISSN: 0219-3116 (cit. on p. 89).

[215] N. Schmidt and A. Lüder, *AutomationML and eClass Integration*, tech. rep., 2015, URL: www.automationml.org (cit. on p. 90).

[216] T. Tudorache, *Employing ontologies for an improved development process in collaborative engineering*, PhD thesis: Technische Universitat Berlin, Fakultat IV, Elektrotechnik und Informatik, 2006 (cit. on p. 93).

[217] R. Mordinyi, D. Winkler, F. J. Ekaputra, M. Wimmer and S. Biffl, "Investigating model slicing capabilities on integrated plant models with AutomationML", *Proceedings of the 21st IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, Berlin, Germany, September 6-9*, 2016 1 (cit. on p. 109).

[218] J. Prinz, "Consistent Merging of AutomationML Documents in Multiple Sources Scenarios", *Proceedings of the 4th AutomationML user conference, Esslingen, Germany, October 18*, 2016 (cit. on pp. 109, 120).

[219] T. Moser, R. Mordinyi and D. Winkler, "Extending mechatronic objects for automation systems engineering in heterogeneous engineering environments", *Proceedings of the 17th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, Krakow, Poland, September 17-21*, 2012 1 (cit. on p. 109).

[220] I. Grangel-González, L. Halilaj, M.-E. Vidal, O. Rana, S. Lohmann, S. Auer and A. W. Müller, "Knowledge Graphs for Semantically Integrating Cyber-Physical Systems", *Proceedings of the 29th International Conference of Database and Expert Systems Applications DEXA, Regensburg, Germany, September 3 - 6*, 2018 (cit. on p. 110).

[221] I. Grangel-González, L. Halilaj, M.-E. Vidal, S. Lohmann, S. Auer and A. W. Müller, "Seamless Integration of Cyber-Physical Systems in Knowledge Graphs", *Proceedings of the 33rd Annual ACM Symposium on Applied Computing, SAC, Pau, France, April 09-13*, ACM, 2018 2000 (cit. on p. 110).

[222] I. Grangel-González, "Semantic Data Integration for Industry 4.0 Standards", *Proceedings of the 20th International Conference of Knowledge Engineering and Knowledge Management EKAW Satellite Events, EKM and Drift-an-LOD, Bologna, Italy, November 19-23, Revised Selected Papers*, 2016 230 (cit. on p. 110).

[223] M. Sabou, F. J. Ekaputra and S. Biffl, "Semantic Web Technologies for Data Integration in Multi-Disciplinary Engineering", *Multi-Disciplinary Engineering for Cyber-Physical Production Systems*, Springer, 2017 301 (cit. on pp. 110, 111).

[224] S. Ceri, G. Gottlob and L. Tanca, *What you Always Wanted to Know About Datalog (And Never Dared to Ask)*, IEEE Transactions on Knowledge and Data Engineering **1** (1989) 146 (cit. on pp. 116, 117, 125).

[225] C. Lange, "Krextor – An Extensible XML→RDF Extraction Framework", *Proceedings of the Workshop Scripting and Development for the Semantic Web (SFSW), CEUR*, 449, 2009 (cit. on p. 117).

[226] M. W. Chekol, G. Pirrò, J. Schoenfisch and H. Stuckenschmidt, "Marrying Uncertainty and Time in Knowledge Graphs", *Proceedings of the 31st AAAI Conference on Artificial Intelligence, California, USA*, 2017 88 (cit. on pp. 118–120).

[227] J. Huber, M. Niepert, J. Noessner, J. Schoenfisch, C. Meilicke and H. Stuckenschmidt, *An infrastructure for probabilistic reasoning with web ontologies*, Semantic Web **8** (2017) 255 (cit. on p. 118).

[228] M. Brettel, N. Friederichsen, M. Keller and M. Rosenberg, *How virtualization, decentralization and network building change the manufacturing landscape: An Industry 4.0 perspective*, International Journal Mechanical, Aerospace, Industrial and Mechatronics Engineering **8** (2014) 37 (cit. on p. 131).

[229] M. Hermann, T. Pentek and B. Otto, "Design Principles for Industrie 4.0 Scenarios", *Proceedings of the 49th Hawaii International Conference on System Sciences (HICSS)*, IEEE, 2016 3928 (cit. on p. 131).

[230] N. Petersen, L. Halilaj, I. Grangel-González, S. Lohmann, C. Lange and S. Auer, "Realizing an RDF-Based Information Model for a Manufacturing Company - A Case Study", *Proceedings of the 16th International Semantic Web Conference ISWC, Vienna, Austria, October 21-25*, 2017 350 (cit. on p. 131).

[231] M.-E. Vidal, K. M. Endris, S. Jozashoori, F. Karim and G. Palma, "Semantic Data Integration of Big Biomedical Data for Supporting of Personalised Medicine", *Current trends on semantic Web Technologies: Theory and Practice*, Springer Verlag on Studies in Computational Intelligence book series, 2017, In Press (cit. on p. 133).

[232] L. Halilaj, I. Grangel-González, M. Vidal, S. Lohmann and S. Auer, "Proactive Prevention of False-Positive Conflicts in Distributed Ontology Development", *Proceedings of the 8th International Joint Conference on Knowledge Discovery, Knowledge Engineering, and Knowledge Management, IC3K, Volume 2: KEOD, Porto, Portugal, November 12-14*, 2016 (cit. on p. 133).

[233] S. Suwanmanee, D. Benslimane, P. Champin and P. Thiran, "Wrapping and Integrating Heterogeneous Relational Data with OWL", *Proceedings of the 7th International Conference on Enterprise Information Systems ICEIS, Miami, USA, May 25-28*, 2005 11 (cit. on p. 137).

[234] H. Wache, T. Voegele, T. isser, H. Stuckenschmidt, H. Schuster, G. Neumann and S. Huebner, "Ontology-based integration of information - a survey of existing approaches", *IJCAI-01 Workshop: Ontologies and Information*, 2001 (cit. on p. 138).

[235] L. Bellatreche and G. Pierra, "OntoAPI: An Ontology-based Data Integration Approach by an a Priori Articulation of Ontologies", *Proceedings of the 18th International Workshop on Database and Expert Systems Applications (DEXA), Regensburg, Germany, September 3-7*, 2007 (cit. on p. 138).

[236] A. Poggi, D. Lembo, D. Calvanese, G. De Giacomo, M. Lenzerini and R. Rosati, *Linking Data to Ontologies*, Journal Data Semantics **10** (2008) 133 (cit. on p. 139).

APPENDIX A

Appendix A

A.1 Publications

During the work on this thesis, some parts of the contributions have been presented at conferences and journals and have been published within the proceedings of those events. In the following these papers are listed:

- *Journal Articles and Book Chapters*:
  1. Lavdim Halilaj, **Irlán Grangel-González**, Gökhan Coskun, Sören Auer. *Git4Voc: Collaborative Vocabulary Development based on Git.* International Journal of Semantic Computing; 10(2): 167-192 (2016).
  2. Olga Kovalenko, **Irlán Grangel-González**, Marta Sabou, Stefan Biffl, Sören Auer, Maria-Esther Vidal. *AutomationML Ontology: Modeling Cyber-Physical Systems for Industry 4.0.* Under review in the Semantic Web Journal.
  3. **Irlán Grangel-González**, Maria-Esther Vidal, Steffen Lohmann, Sören Auer. *Towards a Knowledge Graph of Industry 4.0 Standards.* Under review in the IEEEAccess journal.
  4. Lavdim Halilaj, **Irlán Grangel-González**, Maria-Esther Vidal, Steffen Lohmann, Sören Auer. *SerVCS: Serialization Agnostic Ontology Development in Distributed Settings.* In Springer - Communications in Computer and Information Science (CCIS) - 8th International Joint Conference, IC3K 2016, Porto, Portugal, November 9-11, 2016, Revised Selected Papers.
- *Conference Papers*:
  5. **Irlán Grangel-González**, Lavdim Halilaj, Gökhan Coskun, Sören Auer, Diego Collarana, Michael Hoffmeister. *Towards a Semantic Administrative Shell for Industry 4.0 Components.* In Proceedings of the Tenth IEEE International Conference on Semantic Computing (ICSC) 2016, 230-237, IEEE. Fraunhofer IAIS Paper of the Month, June 2016.
  6. Lavdim Halilaj, **Irlán Grangel-González**, Gökhan Coskun, Sören Auer. *Git4Voc: Git-based Versioning for Collaborative Vocabulary Development.* In Proceedings of the Tenth IEEE International Conference on Semantic Computing 2016, 285-292, IEEE.

7. Niklas Petersen, **Irlán Grangel-González**, Sören Auer, Gökhan Coskun, Marvin Frommhold, Sebastian Tramp, Maxime Lefranc, Antoine Zimmermann. *SCORVoc: Vocabulary-based Information Integration and Exchange in Supply Networks.* In Proceedings of the Tenth IEEE International Conference on Semantic Computing 2016, 132-139, IEEE.
8. **Irlán Grangel-González**, Lavdim Halilaj, Gökhan Coskun, Sören Auer, Diego Collarana. *An RDF-based approach for implementing Industry 4.0 components with Administration Shells.* In Proceedings of the 21st IEEE International Conference on Emerging Technologies and Factory Automation (EFTA) 2016, 1-8, IEEE.
9. Lavdim Halilaj, **Irlán Grangel-González**, María-Esther Vidal, Steffen Lohmann, Sören Auer. *Proactive detection of False-Positive Conflicts in Distributed Ontology Development.* In Proceedings of the International Conference on Knowledge Engineering and Ontology Development (KEOD) 2016, 43-51, SciTePress. *Best Paper Award.*
10. **Irlán Grangel-González**, Diego Collarana Vargas, Lavdim Halilaj, Steffen Lohmann, Christoph Lange, Maria-Esther Vidal, Sören Auer. *Alligator: A Deductive Approach for the Integration of Industry 4.0 Standards.* In Proceedings of the 20th International Conference of Knowledge Engineering and Knowledge Management (EKAW) 2016, 272-287.
11. Lavdim Halilaj, Niklas Petersen, **Irlán Grangel-González**, Christoph Lange, Sören Auer, Gökhan Coskun, Steffen Lohmann. *An Integrated environment to Support Version-Controlled Vocabulary Development.* In Proceedings of the 20th International Conference of Knowledge Engineering and Knowledge Management (EKAW) 2016, 303-319; – Fraunhofer IAIS Paper of the month, April 2017.
12. **Irlán Grangel-González**, Paul Baptista, Lavdim Halilaj, Steffen Lohmann, Maria-Esther Vidal, Christian Mader, Sören Auer. *The Industry 4.0 Standards Landscape from a Semantic Integration Perspective.* In Proceedings of the 21st IEEE International Conference on Emerging Technologies and Factory Automation 2017, 1-8.
13. Niklas Petersen, Lavdim Halilaj, **Irlán Grangel-González**, Steffen Lohmann, Christoph Lange, Sören Auer. *Realizing an RDF-based Information Model for a Manufacturing Company – A Case Study.* (One of the two nominees for the Best In-Use Paper Award) In Proceedings of the 16th International Semantic Web Conference (ISWC) 2017, 350-366, Springer.
14. Lavdim Halilaj, **Irlán Grangel-González**, Steffen Lohmann, Maria-Esther Vidal, Sören Auer: EffTE: a dependency-aware approach for test-driven ontology development, SAC 2018, 1976-1983.
15. **Irlán Grangel-González**, Lavdim Halilaj, Omar Rana, Maria-Esther Vidal, Steffen Lohmann, Sören Auer, Andreas W. Müller. *Knowledge Graphs for Semantically Integrating of Cyber-Physical Systems.* In Proceedings of the 29th International Conference of Database and Expert Systems Applications (DEXA) 2018, 184-199.

- *Short Papers, Workshops and Demos*:

16. **Irlán Grangel-González**, Lavdim Halilaj, Gökhan Coskun, Sören Auer. *Towards Vocabulary Development by Convention.* In Proceedings of the International Confer-

ence on Knowledge Engineering and Ontology Development (KEOD), 2015, 334-343, SciTePress.

17. **Irlán Grangel-González**. *A Semantic Data Integration for Industry 4.0 Standards.* Knowledge Engineering and Knowledge Management – EKAW 2016 Satellite Events, EKM and Drift-an-LOD, 230-237, Springer.
18. Diego Collarana, Mikhail Galkin, Christoph Lange, **Irlán Grangel-González**, Maria-Esther Vidal, Sören Auer. *FuhSen: A Federated Hybrid Search Engine for building a knowledge graph on-demand.* On the Move to Meaningful Internet Systems: OTM 2016 Conferences, 752-761, LNCS.
19. Lavdim Halilaj, Niklas Petersen, **Irlán Grangel-González**, Christoph Lange, Steffen Lohmann, Christian Mader, Sören Auer. *Industrial Data Space: Semantic integration of Enterprise Data with VoCol.* In 12th International Conference on Semantic Systems (Semantics) - European Linked Data Contest, 2016.
20. Lavdim Halilaj, **Irlán Grangel-González**, Maria-Esther Vidal, Steffen Lohmann, Sören Auer. *DemoEffTE: A Demonstrator of Dependency-aware Evaluation of Test Cases over Ontology.* Proceedings of the Posters and Demos Track of the 13th International Conference on Semantic Systems – SEMANTiCS 2017.
21. **Irlán Grangel-González**, Lavdim Halilaj, Maria-Esther Vidal, Steffen Lohmann, Sören Auer, Andreas W. Müller. *Seamless Integration of Cyber-Physical Systems in Knowledge Graphs.* In 33rd ACM/SIGAPP Symposium On Applied Computing (ACM SAC) 2018 Proceedings, SAC 2018, 2000-2003.

List of Figures

List of Tables