Semantic Data Integration for
Supply Chain Management
with a Specific Focus on Applications in the Semiconductor Industry

Der Fakultät für Elektrotechnik und Informatik
der Gottfried Wilhelm Leibniz Universität Hannover
zur Erlangung des akademischen Grades

Doktor-Ingenieurin
(Dr.-Ing.)
genehmigte Dissertation

von Frau
M.Sc. Nour Ramzy
geboren am 03.08.1993
in Kairo, Ägypten

2022

1. Gutachter:	Prof. Dr. Sören Auer
2. Gutachter:	Prof. Dr. Dante Barone
Tag der Promotion:	11.11.2022

Zusammenfassung

Supply Chain Management (SCM) ist für die Überwachung, Steuerung und Verbesserung der Leistung von Supply Chains (SCs) unerlässlich. Die zunehmende Globalisierung und Diversität der SC führt zu komplexen SC-Strukturen, begrenzter Sichtbarkeit zwischen den SC-Partnern und einer schwierigen Zusammenarbeit aufgrund verteilter Datensilos. Die Digitalisierung ist verantwortlich dafür, dass die SCs grundlegender Sektoren wie der Halbleiterindustrie vorangetrieben und umgestaltet werden. Dies wird durch die unvermeidliche Rolle, die Halbleiterprodukte in Elektronik, IoT und Sicherheitssystemen spielen, noch beschleunigt . Halbleiter-SCM ist einzigartig, da die SC Vorgänge besondere Merkmale aufweisen, wie z. B. lange Produktionsvorlaufzeiten und kurze Produktlebensdauer. Daraus folgt, dass systematisches SCM erforderlich ist, um den Informationsaustausch zu etablieren, Ineffizienzen aufgrund von Inkompatibilität zu überwinden und sich an die branchenspezifischen Herausforderungen anzupassen.

Das Semantic Web ist für die Verknüpfung von Daten und den Informationsaustausch konzipiert. Semantische Modelle liefern High-Level Beschreibungen der Domäne, die Interoperabilität ermöglichen. Semantische Daten Integration konsolidiert die heterogenen Daten zu sinnvollen und wertvollen Informationen. Das Hauptziel dieser Arbeit ist es, Semantic Web Technologien (SWT) für SCM zu untersuchen, mit einem speziellen Fokus auf Anwendungen in der Halbleiterindustrie.

Als Teil des SCM gewährleistet die End-to-End (E2E) SC-Modellierung die Sichtbarkeit von SC-Partnern und -Flüssen. Vorhandene Modelle sind in der Art und Weise, wie sie operative SC-Beziehungen jenseits von Eins-zu-eins-Strukturen darstellen, begrenzt. Der Mangel an empirischen Daten von mehreren SC-Partnern erschwert die Analyse der Auswirkungen der Partner des Liefernetzwerks untereinander und das Benchmarking der Gesamtleistung der SC.

In unserer Arbeit untersuchen wir, (i) wie semantische Modelle zur Standardisierung und zum Benchmarking von SCs eingesetzt werden können. Darüber hinaus benötigen SC-Experten in einem volatilen und unvorhersehbaren Umfeld methodische und effiziente Ansätze zur Integration verschiedener Datenquellen, um fundierte Entscheidungen über das Verhalten von SC zu treffen. Daher befasst sich diese Arbeit mit (ii) der Frage, wie semantische Datenintegration dazu beitragen kann, SCs effizienter und widerstandsfähiger zu machen. Um sich außerdem eine gute Position auf einem wettbewerbsorientierten Markt zu sichern, streben Halbleiter-SC danach Strategien zur Kontrolle von Nachfrageschwankungen, d.h. Bullwhip, zu implementieren und gleichzeitig nachhaltige Kundenbeziehungen zu pflegen. Wir untersuchen (iii), wie wir semantische Technologien einsetzen können um speziell Halbleiter-SCs zu unterstützen.

In dieser Arbeit stellen wir semantische Modelle bereit, die auf standardisierte Weise SC-Prozesse, -Strukturen und -Abläufe integrieren, um sowohl ein umfassendes Verständnis der ganzheitlichen SCs zu gewährleisten als auch granulare Betriebsdetails zu enthalten. Wir zeigen, dass diese Modelle die Instanziierung einer synthetischen SC für Benchmarking ermöglichen. Wir tragen mit semantischen Datenintegrationsanwendungen dazu bei, Interoperabilität zu ermöglichen und die SCs effizienter

und widerstandsfähiger zu machen. Außerdem nutzen wir Ontologien und Knowledge-Graphs (KGs), um kundenorientierte Bullwhip-Beherrschungs-Strategien zu implementieren. Wir schaffen semantikbasierte Ansätze, die mit Algorithmen der künstlichen Intelligenz (KI) verknüpft werden, um die Besonderheiten der Halbleiterindustrie zu adressieren und operative Exzellenz zu gewährleisten.

Die Ergebnisse beweisen, dass der Einsatz semantischer Technologien dazu beiträgt, ein rigoroses und systematisches SCM zu erreichen. Wir sind der Meinung, dass eine bessere Standardisierung, Simulation, Benchmarking und Analyse, wie sie in den Beiträgen beschrieben werden, helfen wird, komplexere SC-Szenarien zu meistern. SC-Stakeholder können die Domäne zunehmend verstehen und sind daher besser mit effektiven Kontrollstrategien ausgestattet, um Störungsbeschleuniger, wie den Bullwhip-Effekt, einzudämmen. Im Wesentlichen erschließen die vorgeschlagenen SWT-basierten Strategien das Potenzial, um die Effizienz, Widerstandsfähigkeit und operative Exzellenz von Liefernetzwerken und insbesondere der Halbleiter-SC zu steigern.

Abstract

SCM is essential to monitor, control, and enhance the performance of SCs. Increasing globalization and diversity of SCs lead to complex SC structures, limited visibility among SC partners, and challenging collaboration caused by dispersed data silos. Digitalization is responsible for driving and transforming SCs of fundamental sectors such as the semiconductor industry. This is further accelerated due to the inevitable role that semiconductor products play in electronics, IoT, and security systems. Semiconductor SCM is unique as the SC operations exhibit special features, e.g., long production lead times and short product life. Hence, systematic SCM is required to establish information exchange, overcome inefficiency resulting from incompatibility, and adapt to industry-specific challenges.

The Semantic Web is designed for linking data and establishing information exchange. Semantic models provide high-level descriptions of the domain that enable interoperability. Semantic data integration consolidates the heterogeneous data into meaningful and valuable information. The main goal of this thesis is to investigate Semantic Web Technologies (SWT) for SCM with a specific focus on applications in the semiconductor industry.

As part of SCM, E2E SC modeling ensures visibility of SC partners and flows. Existing models are limited in the way they represent operational SC relationships beyond one-to-one structures. The scarcity of empirical data from multiple SC partners hinders the analysis of the impact of supply network partners on each other and the benchmarking of the overall SC performance. In our work, we investigate **(i)** how semantic models can be used to standardize and benchmark SCs. Moreover, in a volatile and unpredictable environment, SC experts require methodical and efficient approaches to integrate various data sources for informed decision-making regarding SC behavior. Thus, this work addresses **(ii)** how semantic data integration can help make SCs more efficient and resilient. Moreover, to secure a good position in a competitive market, semiconductor SCs strive to implement operational strategies to control demand variation, i.e., bullwhip, while maintaining sustainable relationships with customers. We examine **(iii)** how we can apply semantic technologies to specifically support semiconductor SCs.

In this thesis, we provide semantic models that integrate, in a standardized way, SC processes, structure, and flows, ensuring both an elaborate understanding of the holistic SCs and including granular operational details. We demonstrate that these models enable the instantiation of a synthetic SC for benchmarking. We contribute with semantic data integration applications to enable interoperability and make SCs more efficient and resilient. Moreover, we leverage ontologies and KGs to implement customer-oriented bullwhip-taming strategies. We create semantic-based approaches intertwined with Artificial Intelligence (AI) algorithms to address semiconductor industry specifics and ensure operational excellence.

The results prove that relying on semantic technologies contributes to achieving rigorous and systematic SCM. We deem that better standardization, simulation, benchmarking, and analysis, as

elaborated in the contributions, will help master more complex SC scenarios. SCs stakeholders can increasingly understand the domain and thus are better equipped with effective control strategies to restrain disruption accelerators, such as the bullwhip effect. In essence, the proposed SWT-based strategies unlock the potential to increase the efficiency, resilience, and operational excellence of supply networks and the semiconductor SC in particular.

Semantic Data Integration, Knowledge-graphs, Semiconductor Supply Chain Management

Acknowledgements

This thesis was an uphill journey full of unending series of challenges mixed with countless “yes I did it” moments, and enjoyable events. I wouldn’t have made it this far if it weren’t for some people who were there throughout every step. I am deeply grateful to you. You were and still are my guiding lighthouse, my reliable companions, and my faithful supporters.

I would like to thank my supervisor at Infineon, Hans Ehm. Your guidance, trust, and investment made me grow on a professional and personal level. You’re the world’s best boss.

Prof. Doc. Sören Auer, thank you for always making time for my questions and for welcoming me to your research team. Your wise and professional guidance allowed me to progress with steady steps. Thank you Dr. Javad Chamanara, for always bringing new perspectives to my attention; your thorough and critical comments helped refine the quality of my work.

I also want to thank Patrick Moder and Marco Ratusny for the awesome daily professional companionship. Thank you for sharing the same struggles, frustrations, and fun moments. I am grateful for all the pep talk and paper reviews. My gratitude also goes to Boris Shalumov and Dr. Niklas Petersen, your work set the path for me, and your technical advice has always been a source of inspiration for my work.

My friends, my chosen family, in Germany, Egypt, France, Italy, the USA, Netherlands, and Switzerland, you are my cheerleaders. Thank you for all the fun times we spent together that maintained my sanity in this rough ride. You not only believed in me and encouraged me all the way, but you literally picked me up on the darkest days (with food, drinks, sleepovers, coloring dates, walks, and exercising dates).

Mamy, Papy, Farah, and Shero, your faith and prayers lifted me in my hard days and allowed me to transcend hardships. Words fail in front of your unconditional love and support. You made me who I am.

Liam, you are my anchor. I am forever in debt for your love, good food, wise handling of my stress, fulfilling partnership, countless jokes, and faith in me. I am the luckiest girl to have you as my partner.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Our increasingly globalized economy has resulted in a high interconnectedness between SCs [1]. Consequently, SCs are becoming more dispersed, complex, and diverse. SCs have evolved from being chains of businesses with one-to-one relationships to becoming networks of multiple interdependent businesses that provide products and services to customers [2]. Hence, monitoring and analyzing the behavior of a SC are essential goals of SCM as a slight alteration in performance does not only affect one organization but a highly connected network.

1.1 Motivation

Systematic SCM, observation and control of SC overall performance, has proven fundamental in the recent context of the COVID-19 pandemic, as both the supply and demand sides of SC have been deeply affected [3]. On the supply side, multiple national lockdowns slowed or even temporarily stopped the flow of materials and finished goods, hence disrupting production and manufacturing. The demand increased vastly for some products, e.g., electronics, due to the rising need for personal computers, servers, and equipment, while it decreased significantly for other industries such as the automotive industry.

The misalignment between the surge in global demand and the limited global supply especially influenced the semiconductor manufacturing and led to what is now known as the chip shortage [4]. Furthermore, the bullwhip effect, characterizing the semiconductor domain, emphasized the described supply and demand imbalance by further amplifying the demand in the upstream parts of the SC. Semiconductor SCM is unique in the sense that SC operations display specific characteristics, e.g., long production lead times and short product life, while being at the center of technological advancement and thereby touching almost all industries. Thus, the semiconductor industry requires tailoring SCM strategies to address the complexity of the SC structures caused by the wide range of customers with fluctuating demands for products and rapidly changing technologies in a competitive market.

The aforementioned conditions complicate the operations and reveal the need for SC visibility in order to provide a holistic/comprehensive awareness of the network and to detect the changes, disruptions, and their consequences. Even before COVID-19, events such as natural disasters, transportation blockages, sanctions, etc., shed light on the importance of SC visibility and integration to consistently monitor the behavior and flows, e.g., materials, financial, and information amongst all the SC members. The objective of this thesis is to demonstrate how semantic technologies can enable rigorous SCM for enhanced integration and analysis. We address the semiconductor industry's specific challenges using semantic technologies.

1.2 Unaddressed Supply Chain Management Challenges

Based on the presented motivation to establish comprehensive SCM in order to monitor and guide SCs, we list the entailed challenges. We derive the first three challenges as untackled general SC hurdles by literature and existing SCM approaches. Given the relevance of the semiconductor industry, we rely on the industrial context of this thesis to identify the fourth and fifth domain-specific problems.

Challenge 1: Create standardized SC models Integrated modeling is required for visibility and proactive monitoring of members and processes across the SC network [5]. Recent works have established SC models incorporating core relations and structures. However, such models are still rather isolated, thus preventing a holistic view of the SC. Existing SC models created by one organization are limited in the way they grasp the dynamics between SC partners beyond their one-to-one 'dyadic' relationships. They are not extensive enough to incorporate an E2E SC view while also including standard operational SC artifacts. Semantic modeling provides high-level descriptions of the domain to integrate SC pillars and increase understandability. Here, we identify the need and present initial comprehensive semantic E2E SC models that rely on existing standards to integrate partners, flows, operations and processes. In fact, SC models mimic reality and provide the means to simulate and benchmark the overall performance under multiple empirical scenarios.

Challenge 2: Benchmark SC performance Given the competitive trait of SCs, it is of essential importance to compare and benchmark SC behavior, consequently triggering learning outcomes and improvements [6]. We identify a lack of E2E SC data that enables integrated analysis of the SC. Existing logs or data from one company are not enough to validate the E2E SC models. Thus, we tackle the challenge of benchmarking the performance of an E2E SC. Moreover, the modeling and analysis of large SCs are computationally intensive and require extensive amounts of data. Hence, we address via semantic modeling the necessity to establish competent ways to create E2E SC models for benchmarking and analysis. Namely, SC benchmarking benefits from the integration of various distributed data sources, thus, the next step is to address SC data integration.

Challenge 3: Establish data integration and information exchange for more efficiency and resilience As part of SCM, SC experts and stakeholders are required to make critical and urgent decisions. Due to the complexity and globalization of the environment, SC generates siloed and dispersed data sets which prevents the decision-making process from being methodical and efficient. Data integration enables SC stakeholders to make informed decisions regarding SC structure and operational strategies during complicated situations. Therefore, we address the need to overcome SC dispersed data sources using semantic data integration by establishing agile information exchange to make SCs more efficient and resilient.

Challenge 4: Create customer-oriented bullwhip-taming strategies for semiconductor SC To be successful in the semiconductor domain, companies need to control high demand volatility and the bullwhip effect. Companies rely on various strategies, e.g., forecasting and dynamic pricing, to achieve smooth demand planning, efficient capacity utilization, and minimization of inventory while generating revenue. Nevertheless, some of these strategies risk harming the relationships with the customers. We identify the necessity to create approaches to address semiconductor SC challenges such as the bullwhip while maintaining close associations with customers to identify their specific needs.

Challenge 5: Ensure good data quality for operational excellence in semiconductor SCs Ensuring good data quality enables semiconductor SCs to maintain operational excellence and avoid errors that might lead to customer dissatisfaction. Recovering from missing and incorrect values is essential yet a cumbersome task, especially with complicated domain knowledge such as in the semiconductor SCs. We demonstrate that capturing domain knowledge using semantic models ensures good data quality, that in turn, enhances the performance of a machine learning prediction model for SC applications.

Semantic technologies, i.e., semantic modeling and semantic data integration, comprehensively capture core artifacts to increase the understandability and control of the domain. We rely on semantic technologies to establish systematic SCM, address the identified SC challenges, and tackle semiconductor domain-specific hurdles. This endeavor unfolds into the following research questions.

1.3 Research Questions

To address the problems described previously, we define the following research questions:

> RQ1: How can semantic models be used to standardize and benchmark supply chains?

E2E SC modeling mimics the real-world and allows an empirical and coupled environment to study and control complex and dispersed SCs. Semantic modeling especially provides coherent and high-level descriptions, which ensure a standardized representation of the domain they illustrate. Thus, we use semantic models to capture core concepts of the E2E SC environment in a standardized way. We integrate existing SC models using semantic artifacts to facilitate the analysis of the E2E SC interactions, collective behavior, and operational performance. We provide methods leveraging ontologies and KGs to overcome the scarcity of integrated E2E SC data. Additionally, we propose approaches that rely on semantic models to create empirically controlled setups to simulate the E2E SC behavior. Based on these contributions, we demonstrate the impact of using semantic models to standardize and benchmark SCs.

> RQ2: How can semantic data integration help make supply chains more efficient and resilient?

Semantic models provide a comprehensive description of the domain, which allows SC information integration and exchange. We implement semantic data integration applications that leverage semantic models to reach interoperability and make SCs more efficient and resilient. While production efficiency is not the primary scope of this thesis, instead, we propose Master Data (MD) applications. We propose a knowledge-graph-based Master Data Management (MDM) implementation that integrates MD data perspectives from various SC stakeholders to create a unified view of the SC MD and models. We demonstrate the potential of relying on semantic data integration to ensure efficiency in SC reporting and decision-making.

Moreover, we tackle semantic Disruption Management Process (DMP) to increase the resilience of supply networks by allowing SC data integration. We show how semantic models, while resembling SC digital twins, facilitate an optimized control in complex SC scenarios and enable integration to discover the link between SC disruption and performance deterioration. For MDM and DMP, we prove that semantic artifacts implement data integration applications while being aligned with classical non-semantic SC applications.

> RQ3: How can we apply semantic technologies to specifically support semiconductor supply chains?

In analyzing how to support the semiconductor SCs, we create Semantic Web (SW) applications to optimize the utilization of production capacities and planning while accommodating customers' unstable ordering behavior. We address the issue of long production times and demand volatility and propose semantic data integration to create centralized information that enables customer-tailored revenue management strategies.

Moreover, understanding, classifying, and predicting the Customer Order Behavior (COB) are key to taming the demand variability. Thus, we demonstrate the benefit of comprehensive representation of the domain, provided as input by semantic models, on the performance of a Machine Learning (ML) algorithm classifying the COB. Additionally, in a highly distributed SC, ensuring good quality of the operational data is essential to avoid problems in customs declaration while delivering to various countries. We analyze the benefit of SWT on enhancing the quality, i.e., correctness and completeness of SC data. Then, we show the effect of the corrected and complete values for better prediction of semiconductors' packing information via ML algorithm. We establish the business impacts and implications of our implementations on the semiconductor industry.

1.4 Industrial Context

We conducted this research at Infineon Technologies AG. This industrial context shaped the problems and provided the industry-specific challenges we address.

1.4.1 Infineon Technologies AG

Infineon is a German semiconductor manufacturer founded on 1 April 1999. Infineon's motto is the commitment to making life easier, safer, and greener. It offers semiconductors and systems for automotive, industrial, and multimarket sectors, as well as chip card and security products. Infineon is split into several divisions, i.e., Automotive (ATV), Industrial Power Control (IPC), Power and Sensor Systems (PSS), and Connected Secure Systems (CSS). Each division is linked to some organization units, i.e., product and business lines; every product line has several associated products.

Driven by preemptive digitalization, Infineon acknowledges that humans and machines are producing enormous amounts of data and that Big Data is an extremely valuable resource [7]. Consequently, Infineon supports research aiming to cope with the increasing data volume and complexity while catering to the semiconductor industry's specific needs. The goal of the Corporate Supply Chain

Innovation (CSC IN) team, where this research was conducted, is to address SC challenges relying on innovative technologies, e.g., Semantic Web.

1.4.2 Funded Projects

The shift towards Industry 4.0 paradigm prompted Infineon to communicate and network with various partners to collaborate and share knowledge and expertise about digitization through collaborative projects and consortia. Infineon joined several projects by the public-private Electronic Components and Systems for European Leadership Joint Undertaking (ECSEL JU). ECSEL JU supports funded research, development, and innovation projects in key enabling technologies, e.g., electronics, for the European industry. This program includes small and medium-sized enterprises (SMEs) from various research and technology organizations based in the European Union. Working on the following European-funded projects inspired the use cases implemented in this thesis. The exchanges with project partners from different academic and industrial backgrounds influenced the development and maturity of our methodologies.

Productive4.0 (May 2017 - April 2020) [1]: Productive4.0 is an ECSEL project with the mission to establish a link between the real and digital world by efficiently designing and integrating both the hard- and software of Internet of Things (IoT) devices. The main objective of Productive4.0 is to improve the digitalization of the European industry by creating a user platform across value chains and industries, thus promoting the digital networking of manufacturing companies, production machines, and products. Productive4.0 focuses on generating digital twins for the SCs relying on different technologies, e.g., SW and simulation models. Within Productive4.0, work package 7, we introduce the Digital Reference (DR) ontology as an SC semantic model mirroring the semiconductor SCs, (cf. section 4.2 for details)

Semantically Connected Semiconductor Supply Chains (SC^3) (October 2020 - September 2023) [2]: SC^3 is a unique ECSEL "Communication and Support Action" project. SC^3 creates a generic industrial reference platform for collaborative ontology development, which aims to evolve as an open standard and a basis for a commercial Business2Business platform. This project enables seamless data collaboration among semiconductor companies, SCs containing semiconductors and other industrial domains, e.g., automotive and pharmaceutical industries. Within SC^3, we establish a process to extend and maintain the DR with the required domain ontologies.

Integrated Development 4.0 project (idev40)(May 2018 - October 2021) [3]: idev40 leads the digital transformation towards an integrated digital value chain based on the "digital twin" concept. Many core areas like virtual manufacturing, experiment control, remote development, and dynamic pricing are addressed. We develop KnowGraph-PM, a knowledge-graph-based pricing model relying on the dynamic pricing and revenue management approaches developed within idev40 (cf.section 6.1 for more details).

[1] https://productive40.eu/
[2] https://sc3-project.automotive.oth-aw.de/
[3] http://www.idev40.eu/

Cognitive Economy Intelligence Platform for the Resilience of Economic Ecosystems (CoyPu)(June 2021 - May 2024) [4]: CoyPu (grant 01MK21007A) is a funded project within the program Federal Ministry for Economic Affairs and Climate Action (BMWK). CoyPu aims to develop a platform to integrate, structure, connect and analyze heterogeneous data from supply networks as well as the industry environment and social context. This includes configurable dashboards and tools for AI analysis that provide decision-makers in politics and business with reliable, up-to-date decision-making resources and recommendations for managing crises and achieving increased resiliency. We develop SENS and SENS-GEN in the context of CoyPu.

1.5 Thesis Overview

The thesis is structured to answer the research questions and address the described motivation and challenges. In the following part, we give an overview of the thesis, including the research map and the list of contributions and publications.

1.5.1 Research Map

In this thesis, we rely different artifacts of SW, i.e., Semantic Modeling, Semantic Data Integration, and Semantic Web for Artificial Intelligence to enable SC modeling and applications. We show in Figure 1.1 an overview of our contributions.

We leverage *Semantic Modeling* to conceptualize SCs and describe the SC domain. We propose the following contributions: SENS is a conceptual model for E2E SC, used by a data generator (SENS-GEN) to create synthetic SC instances. We define a semantic reference model for the semiconductor domain: the DR. We implement an ontology-based simulation modeling approach. We apply the proposed methodology to a use case from the semiconductor domain. These contributions overlap as DR incorporates SENS model and is used as the ontology input for the ontology-based simulation approach. With the previous contributions, we address challenges 1 and 2 to create an E2E SC model and benchmark its performance.

We use *Semantic Data Integration* for SC MDM and DMP. Namely, KnowGraph-MDM and MARE address challenge 3 to establish data integration and information exchange for more efficiency and resilience. Furthermore, KnowGraph-PM is a knowledge-graph-based dynamic pricing model that relies on semantic data integration to create a customer-oriented bullwhip-taming strategy for semiconductor SCs, i.e., challenge 4.

We contribute with two research papers that use *Semantic Web for Artificial Intelligence* for semiconductor SC applications. We present SCIM-NN, a Semantic Customer Order Behavior Classification, to control the bullwhip effect. Furthermore, we implement an application for an ontology-based preprocessing to predict product packing details to overcome challenge 5.

SENS, MARE and KnowGraph-MDM are domain-agnostic contributions (located on the left side of the figure) and belong to the *Semantic Supply Chain Management* research area. As we move to the right side (towards *Semantic Semiconductor Supply Chain Management*), the Digital Reference, SCIM-NN and ontology-based preprocessing are driven by the semiconductor industry's challenges, thus considered domain-centric contributions. KnowGraph-PM and the ontology-based simulation work provide a general approach for SC but applied in use cases from the semiconductor domain.

[4] `https://www.coypu.org//`

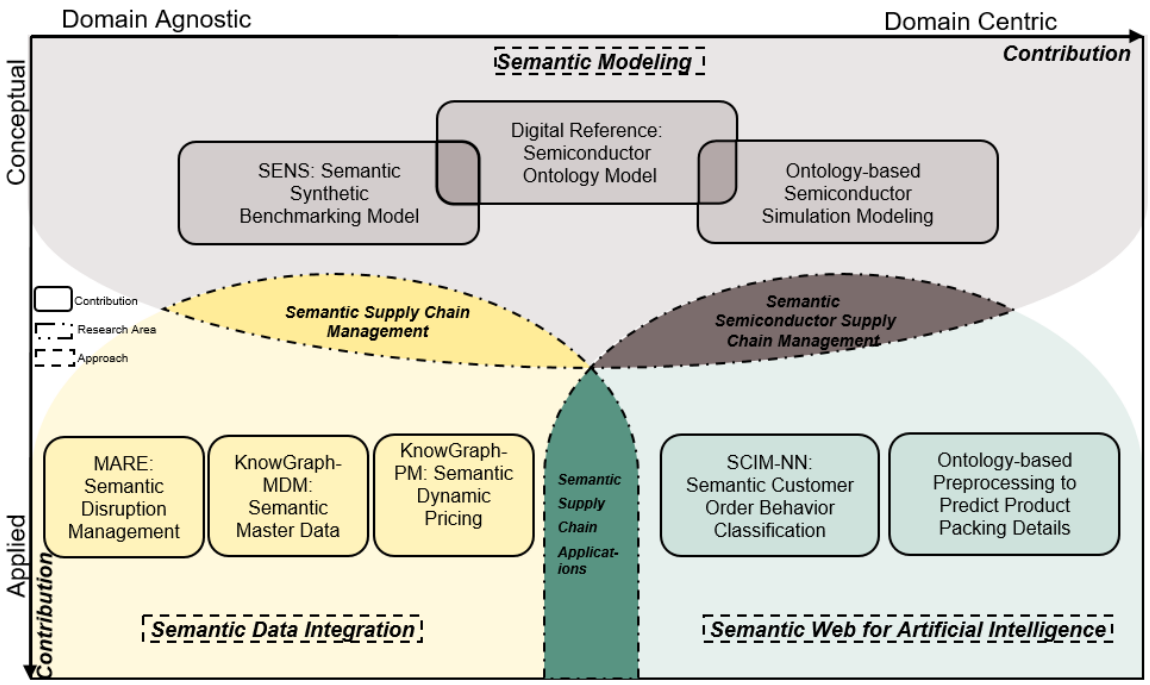


Figure 1.1: Overview of our main contributions and publications listed in subsection 1.5.3. The horizontal direction indicates the move from domain-agnostic to domain-centric contributions. The vertical direction shows the transition from conceptual to applied contributions.

Thus, we position them in the middle of the figure. The contributions in the upper part of the figure are more conceptual than the applications-oriented work at the bottom part, described by the *Semantic Supply Chain Applications* research area.

1.5.2 Contributions

The contributions of this thesis are diverse as they include semantic models, methodologies, use case-specific frameworks to apply semantic technologies for SCM modeling and applications.

- Semantic models (ontologies) for the SCM
  - **SENS** is a standardized integrated semantic model that provides an overall view of SCOR E2E SC structure and flows. This vocabulary is used to generate synthetic SC data compensating for the scarcity of the overall benchmarking data via **SENS-GEN**. The methodology to use SENS and SENS-GEN is in section 4.1.
  - **Supply and Demand Vocabulary** models the demand as orders of products and corresponding details as well as capacity and production information of suppliers. We present a SPARQL-based demand fulfillment algorithm that relies on this vocabulary to simulate SC production planning and scheduling. SPARQL-based performance indicators can measure an empirical SC behavior. We detail in section 4.1 the methodology to apply this vocabulary.

- **Digital Reference (DR) Vocabulary** is a standardized vocabulary for semiconductor SCs. DR is publicly available on `https://w3id.org/ecsel-dr`. We introduced the DR in Productive4.0 project and we maintain and extend it during (SC^3) project. In section 4.2, we describe the details of DR and the included sub-ontologies.
- **Master Data Ontology** models core Master Data components. In **KnowGraph-MDM** presented in section 5.1, we instantiate MD ontology to create a MD KG subsuming MD objects while including various SC stakeholders' perspectives in the model. This contribution enables integrated and efficient SC analysis, reporting and decision-making for MD.
- **Disruption Ontology** describes the disruptive events affecting the SC. Via MARE, we instantiate the disruption ontology to create a disruption KG, i.e., a specific instance of a disruption event. This contribution enables disruption monitoring and determining suitable recovery strategies for this event. In section 5.2, we describe the application of this vocabulary in MARE.
- **Context Information Ontology** contains context information of a customer in the SC and the corresponding temporal granularity. In SCIM-NN (cf. section 6.2), we map the context information to the classes in the created ontology and create the context KG. We use the generated KG embeddings as a second input to the multi-stream neural network to predict customer order behavior.
- **Packing Information Ontology** focuses on the connection between the semiconductor products and their specific packing as shown in section 6.3. Ontology-based reasoning validates the packing information and ensures high-quality data based on the constraints defined in this ontology. This vocabulary contributes as part of the data cleaning in the preprocessing stage of an AI algorithm and supports the selection of features to increase the prediction model performance.

- **SENS-GEN** is a highly configurable data generator that leverages the integrated semantic SC model to produce exemplary data based on input parameters and create a specific synthetic integrated instance of a Supply Chain Network (SCN). The detailed code is published as a technical documentation report [8] and detailed in Appendix B.

- **MARE** is an evaluation framework to simulate the behavior of a synthetic SC under various exemplary disrupted events and an evaluation framework to analyze recovery performance. The detailed code and the technical documentation report are published on [9] and described in Appendix D. To ensure and enhance SC resilience, SC stakeholders can rely on the DMP and resilience evaluation framework in MARE to extract decisions regarding SC structure and operational strategies.

- Semantic data integration applications and evaluation in the domain of SC, i.e., **KnowGraph-MDM** and **MARE**. We present **KnowGraph-PM** a semantic-based pricing solution for revenue management and customer relationship management in the semiconductor domain.

- Methodology for using semantic models to standardize simulation model creation for SCs, cf. section 4.3. Methodologies and use cases for the SW models as an enabler for AI models cf. section 6.3 and section 6.2.

- Active role in the proposal writing of *Semantically Connected Semiconductor Supply Chains (SC^3)*. We also contributed to the writing and submissions of technical reports and milestones documents, and deliverables for the funded projects mentioned in the previous section.
- Supervision of master thesis of students from the CSC IN team at Infineon that led to the following contributions *An ontology-based approach for preprocessing in machine learning: use case for packing material information* and *SCIM-NN: Semantic Context Information modeling for Neural Networks in Customer Order Behavior Classification*

1.5.3 List of Publications

Parts of the work presented in this thesis have already been published/submitted as conference and workshop articles. In the following, the main publications building the basis of this thesis are outlined based on order of appearance in the upcoming chapters. A complete list of publications completed during the Ph.D. term is available in Appendix A.

- **SENS: Semantic Synthetic Benchmarking Model for Integrated Supply Chain Simulation and Analysis** Nour Ramzy, Sören Auer, Javad Chamanara, Hans Ehm. In Proceedings of the *30th European Conference on Information System*, 2022.
- **Digital Reference–A Semantic Web for Semiconductor Manufacturing and Supply Chains Containing Semiconductors** Hans Ehm, Nour Ramzy, Patrick Moder, Christoph Summerer, Simone Fetz, and Cédric Neau. In Proceedings of the *2019 Winter Simulation Conference (WSC). IEEE*, 2019.
- **First Steps Towards Bridging Simulation And Ontology To Ease The Model Creation On The Example Of Semiconductor Industry** Nour Ramzy, Christian James Martens, Shreya Singh, Thomas Ponsignon, and Hans Ehm. In Proceedings of the *2020 Winter Simulation Conference (WSC). IEEE*, 2020.
- **KnowGraph-MDM: A Methodology for Knowledge-Graph-based Master Data Management** Nour Ramzy, Sandra Durst Martin Schreiber, Sören Auer. Submitted in the workshop proceedings in the *24th IEEE International Conference on Business Informatics*, 2022.
- **MARE: Semantic Supply Chain Disruption Management and Resilience Evaluation Framework** Nour Ramzy, Sören Auer, Javad Chamanara, Hans Ehm. In Proceedings of the *24th International Conference on Enterprise Information Systems (ICEIS)*, 2022.
- **KnowGraph-PM: A Knowledge-Graph-based Pricing Model for Semiconductor Supply Chains** Nour Ramzy, Sören Auer, Javad Chamanara, Hans Ehm. In Proceedings of the *8th International Conference on Computational Science/Intelligence, Applied Informatics. Springer*, 2021.
- **SCIM-NN: Semantic Context Information modeling for Neural Networks in Customer Order Behavior Classification**. Philipp Ulrich, Nour Ramzy, Marco Ratusny. Submitted in Proceedings of the *IEEE Transactions on Semiconductor Manufacturing, Special Issue on Production-Level Artificial Intelligence Applications in Semiconductor Manufacturing,* 2022.

- **An ontology-based approach for preprocessing in machine learning: use case for packing material information**. Patricia Centeno Soto, Nour Ramzy, Felix Ocker, Birgit Vogel-Heuser. In Proceedings of the *25th IEEE International Conference on Intelligent Engineering Systems*, 2021.

1.6 Thesis Structure

After the introductive chapter above detailing the industrial context, research questions, and contributions, this dissertation proceeds in the following chapters:

- **Chapter 2** presents the fundamental concepts about SCs modeling and SCM. We examine existing SC models and the included artifacts. We dive into core concepts SCM, e.g., MDM and DMP. We explain the semiconductor industry and the entailed characteristics of semiconductor SCs.
- **Chapter 3** provides background knowledge about SW and corresponding standards. We present semantic SC models. We discuss the related work and existing semantic applications for SCM. From this, we derive scientific gaps that motivate our work.
- **Chapter 4** analyzes the semantic technologies to standardize and benchmark SCs. First, we introduce an E2E standard semantic SC model. We propose a highly configurable data generator that leverages an integrated semantic model of core SC concepts. Consequently, we create synthetic semantic SC data under various scenario configurations for comprehensive analysis and benchmarking applications. Second, we introduce the DR, a semantic vocabulary that serves as a standard for semiconductor SCs. Besides, we highlight how the DR supports creating benchmark simulation models to analyze SC performance. This chapter answers **RQ1**.
- **Chapter 5** focuses on semantic data integration SC applications. We propose a methodology for a knowledge-graph-based MDM which relies on establishing a KG layer for efficient reporting and decision-making. We present MARE, a semantic disruption management and resilience evaluation framework, to integrate data covered by all DMP steps. This chapter answers **RQ2**.
- **Chapter 6** describes industry-specific applications. We present semantic applications to overcome semiconductor SCs challenges. Also, we demonstrate how semantic models serve as an enabler for ML models to address semiconductor domain challenges. This chapter answers **RQ3**.
- **Chapter 7** concludes the thesis with a discussion on the results and a reflection on the answers to the research question. We discuss the outlook and implications of our work.

CHAPTER 2

Supply Chain Preliminaries

In this chapter, we give preliminary knowledge about Supply Chain Management concepts. We provide an overview of SC modeling, Master Data Management, and Disruption Management Process. Then, we present the semiconductor domain and the characteristics of the semiconductor SCs.

2.1 Supply Chain Management

In this section, we introduce SC models that incorporate core aspects. Also, we present MDM to handle SC MD and create a consistent definition of business entities (customer, product, location) to enable efficient, integrated data reporting and analysis. Additionally, we examine DMP to ensure SC resilience and preparedness against disruptive events.

2.1.1 Supply Chain Modeling

SC modeling represents the real-world and creates an empirical, coupled domain to study and monitor SCs. SC models incorporate static and dynamic, structural, and behavioral aspects of SCs

Supply Chain Standard Models

We review the SC concepts incorporated by Supply Chain Operations Reference (SCOR) and E2E SCN as they address important aspects, such as standardization and structural coherence.

SCOR Model. To evaluate SC performance and continuously improve upon it, SC standardization offers a mutual understanding of concepts and processes, consequently enabling benchmarking and comparison of performance. The classic SCOR model, introduced by APICS[1] in 1997, provides a common terminology to define SC standardized activities and performances [10].

The SCOR model covers all customer interactions (order entry through paid invoice); we refer to this as **(C1)**. Additionally, it spans all physical material transactions **(C2)** and all market interactions (from the understanding of aggregate demand to the fulfillment of each order) **(C3)**. Also, the SCOR model contains standard descriptions of the SC processes, e.g., *Source*, *Plan*, *Make*, *Deliver*, *Enable* and *Return*, **(C4)**. Furthermore, the SCOR model organizes SC performance metrics, i.e.,

[1] https://www.ascm.org/

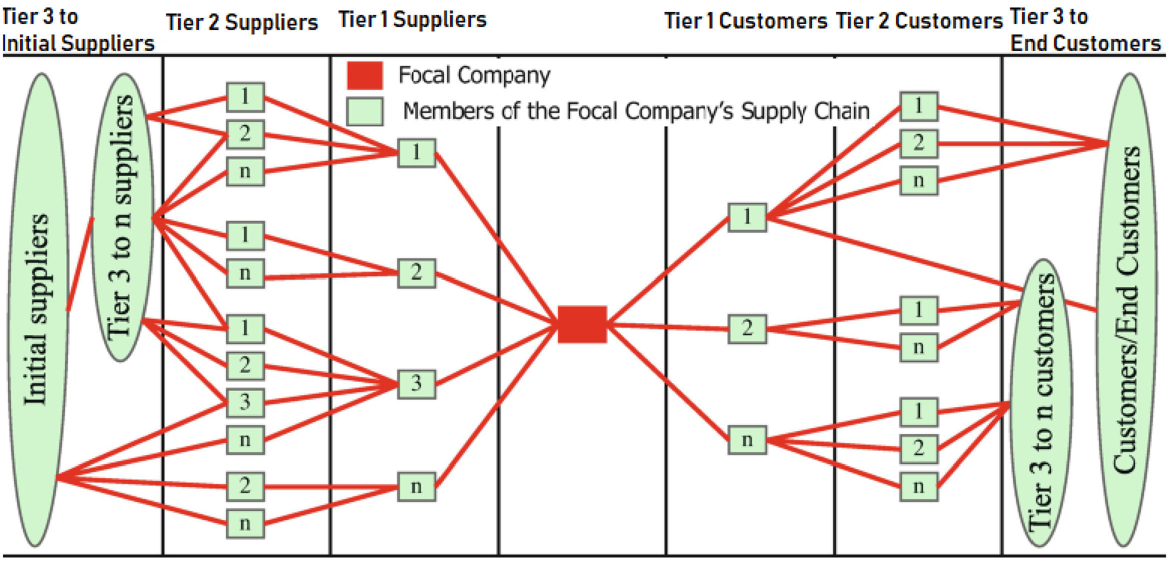


Figure 2.1: Supply chain structure with network topology, nodes as SC partners, and links.

Key Performance Indicator (KPI), into a hierarchical structure **(C5)** to determine and compare the performance of SC on various levels, e.g., top strategies, tactical configurations and operational processes [11]. In addition, SCOR describes best-in-class management practices **(C6)** and maps software products that enable best practices **(C7)**. In order to gain an overall perspective of SC operational performance and structural coherence, E2E SCN models are fundamental.

End-to-End Network Model. An SCN is a network representation of the physical nodes of an SC and how they relate to one another [12]. The E2E model provides an overall perspective of the SC nodes topology that starts at the procurement of raw materials and ends at the delivery of finished goods to the end customers. The literature review by [13] highlights key SC aspects in an E2E SCN model. The authors identify that an SCN consists of a representation of vertices, i.e., nodes, representing SC partners **(C8)**. SC partners are connected with edges **(C9)** modeling product, demand flow, and contractual relations as shown in Figure 2.1 modeled by [2]. Nodes are organized in tiers, nodes in the same tier supply goods and services for the following tiers.

An SCN model considers various materials used to manufacture the end product **(C10)**. The authors describe that the focal company, i.e., Original Equipment Manufacturer (OEM), distinguishes between supply and demand flows, i.e., **(C11)**. Partners in the SCN can be facilities, companies, or warehouses. Nevertheless, the competition in the future will be SC vs. SC where each node participates in one or more SCs **(C12)** while sharing and competing with other nodes over suppliers and customers [14].

Gap Analysis for SC Models. We examine the literature reviews by [15], [13] for existing SCOR and E2E models, respectively. We identify the gap between the artifacts in the studied models and the previously listed SC aspects **(C1-12)** summarized in Table 2.1. We note that existing SC SCOR models do not include management practices and software products **(C6, C7)** as they are considered sensitive information, in order to keep a competitive advantage [15]. Moreover, [13] creates a comparison framework of SC E2E network models and concludes that the academic literature

Model	Abbreviation	Supply Chain Aspect
SCOR	**C1**	Span all customer interactions
	C2	Span all physical material transactions
	C3	Show all market interactions
	C4	Contain standard descriptions of the process
	C5	Represent the SCOR metrics
	C6	Describe best-in-class management practices
	C7	Map of software products for best practices
End-to-End	**C8**	Represent vertices
	C9	Represent edges
	C10	Consider various materials
	C11	Distinguish supply, demand
	C12	Represent SC vs SC

Table 2.1: Supply chain core concepts covered by SCOR and End-to-End models and the abbreviation codes, e.g., C1,C2.

does not contain studies that addresses the topology of SCN **(C8, C9)** together with detailed insights on operational information **(C10, C11)**. Additionally, emergent SCN topology literature address SC node operations independently and not as part of one or many SCs **(C12)** [16].

In an attempt to fulfill the shortcomings of existing models, we study hybrid models that integrate SC aspects from various underlying base models. We consider models that combine SCOR and E2E SCN and the corresponding SC concepts, i.e., **(C1-7), (C8-12)**. Namely, the model by Xiao et al. [17] include SCOR metrics **(C5)** and various raw materials **(C10)** while modeling the SCN, subsuming vertices, edges, various materials, and supply and demand **(C8-11)**. Also, the work by Huan et al [18] model the SCOR process descriptions and metrics **(C4, C5)** while including SC partners as vertices and corresponding relationships as edges **(C8, C9)**. However, existing models do not cover the following SCOR notions: customer interactions, material transactions, market interactions, management practices, and software products. There exist other modeling approaches such as simulations and AI models that mimic the SC behavior and the interactions of its partners, describe SC scenarios, and support decision-making.

Simulation Supply Chain Models

With simulation models, SC behavior can be better understood, and its performance can be empirically assessed with 'what-if' scenarios [19]. Consequently, decision-making in SCM relies on simulation models to represent the real environment by offering a risk-free and flexible virtual world. Hence, we can study expensive and implausible changes in the real world using the virtual world. There exist several simulation types to model the SC: system dynamics (SD), discrete-event dynamic systems (DEDS), and business games. An SD model views SCs as systems with six types of flows, namely, materials, goods, personnel, money, orders, and information [20]. The second represents individual events as well as uncertainties. Also, business games are used to model human operational behavior in a simulated world that represents the SC.

We summarize in Table 2.2 the difference between SD and DES. We notice that SD models are on a high level of granularity, whereas DES describes simulated domains on a low level of granularity.

Criteria	System Dynamics (SD)	Discrete Event (DE)
Levels of Aggregation	High	Low
Data Requirements	Quantitative and Qualitative	Quantitative from processes
Construct Behavior change	Yes	No
Types of Modeling Procedure	Top-down	Bottom-Up
Models Complexity	Low	High
Time Advance Mechanisms	Time Step	Next Event
Feedback Effects	Closed loops structure	Open loops structures
Nature of Problems Modelled	Strategic	Tactical, Operational
Nature of the model	Deterministic	Stochastic
User perception of the model	Transparent	Opaque

Table 2.2: Comparison between system dynamics (SD) and discrete event simulation (DES) models .

Consequently, the nature of the problems modeled and the model complexity are different. SD models are of low complexity and tackle strategic problems that require an overall view of the domain, while DES models are more complex and solve tactical and operational problems requiring a high level of detail. Also, temporal mechanisms in SD models are time steps; the model is deterministic and transparent while having a construct that changes with execution and is receptive to feedback loops. Yet, the DES models advance in time based on the next event; they are stochastic and blackbox. The structure of a DES model does not change during execution and subsumes open loops. Creating an SD model requires a mixture of quantitative and qualitative data and is considered a top-down modeling approach, while DE modeling is bottom-up and requires a high amount of quantitative data. Simulation models mimic the SC structure and behavior for analysis purposes. Similarly, SCM solutions based on artificial intelligence (AI) are expected to be effective instruments to help organizations tackle various challenges.

Artificial Intelligence Supply Chain Models

AI improves human decision-making processes and the entailed productivity in business purposes as it allows us to recognize business patterns, learn business phenomena, seek information, and analyze data intelligently [21]. In a literature review by [22], the authors outline the most prominent AI techniques used in SCM. For instance, techniques such as Artificial Neural Networks (ANN), Expert Systems (ES), data mining, and decision trees are used in SCM applications, e.g., sales forecasting, marketing, pricing and customer segmentation, production forecasting, supplier selection, demand management, and consumption forecasting. [23] presents AI-based business models of different case companies. [24] applies ES and ANN for supplier selection, i.e., to minimize supplier risk and forecast the credit risk in SC finance. [25] relies on reinforcement learning and ANN for spare part demand forecasting and inventory control.

Machine learning models potentially rely on existing data generated by the SC. Data consistency and completeness highly impacts the performance of the model. Consolidation and integration of data are essential for analysis, reporting, and planning [26]. Among SC departments and stakeholders, MD is defined as the “Golden Record” of data where MDM handles MD to maintain and control the growing complexity.

2.1.2 Master Data Management

[27] explains that the view of SCs is based on internal data and seemingly relies on siloed or outdated data sets. Master Data (MD) aims at a consistent definition of business entities (customer, product, location) and corresponding data integration across multiple systems [28]. In MDM, business and IT work together to create the link between physical and conceptual models of the enterprise's MD. Consequently, MDM provides access to accurate information which enables integrated MD reporting and analysis. Also, MDM creates a common language for MD across an SC [29], [30] and synchronizes different stakeholders' inputs and perspectives on the company's data. Creating synchronized and unified definitions of business models and entities supports the decision-making process in the SC. Stakeholders rely on the 'Golden Record' of the data , i.e., MD to access, methodically monitor, and efficiently report on SC data.

Master Data Requirements

We identify two requirements for MDM that we deem necessary and not sufficiently addressed by state-of-the-art research and approaches.

Master Data Modeling and Reporting (R1). MD architecture comprises conceptual and data models. First, the conceptual model describes a company's key business objects and their relationships on a schematic level. This indicates the schema as an abstract overview of the information structure from a business perspective [31]. Second, the data model consists of an application architecture containing the entirety of a company's applications that create, store, and update instances coming from various data formats (structured and unstructured) [32].

Reporting is the creation of reports to access and analyze different MD aspects. Reporting enables SC partners to make informative decisions regarding business-entities modeling and data structures. It allows understanding the conceptual model, retrieving data from the physical layer and connecting/mapping the conceptual model to the underlying data. Reporting tools such as Excel and Tableau access the data from data applications, i.e., data warehouses, and publish it [33]. Various business-related analyses entail different MD reporting requirements. For example, some data reports contain technical product information, whereas others deliver information about facility routes. Consequently, many reports are generated and characterized by duplication and redundancies. The understanding of the conceptual and data models as well as the relations between them is necessary to standardize and consolidate MD reporting and create efficient and methodological decision-making process [34]. Thus, we refer to **R1** as the need for mapping and linking between conceptual and data models to create an integrated reporting scheme for MD.

Stakeholders Involvement (R2). MD is a collaborative discipline that involves interaction between different parties, e.g., MDM experts, business stakeholders, and domain experts as well as sales and marketing participants [35], [36], [37]. MDM experts are responsible for exchanging and synchronizing the reference model across systems, so they consolidate various definitions from stakeholders to create the MD as a single source of truth. Domain experts have the expertise for the domain in question. Therefore, their definitions of MD models are confined to technical and specific requirements, such as details about engineering processes or product descriptions. Also, business stakeholders define MDM from a business requirements viewpoint, namely, the compliance of the

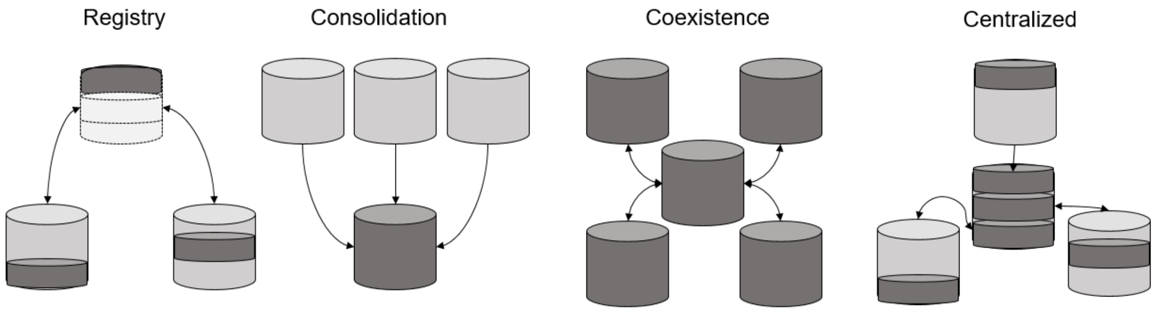


Figure 2.2: Master data management implementation styles: registry, consolidation, coexistence, and centralized.

models to the company's strategies, visions, and goals. Further, sales and marketing parties define MD to suit respective strategies, e.g., online sales name for a product or different pricing models based on market segmentation. MDM ensures the synchronization of different stakeholders' inputs, i.e., **R2** which creates a unified view of MD.

Related Work and Gap Analysis

We examine the extent to which existing MDM approaches, such as the registry, consolidated, coexistence, and centralized implementation styles [38], meet the requirements. We show in Figure 2.2, adapted from [37], the differences between the implementation styles.

First, the *registry style* provides a read-only source of MD which minimizes data redundancy by assigning unique identifiers to matched MD business objects [38], [39], [40]. Second, the *consolidation style* integrates MD from multiple sources into a single managed MDM hub for reporting and reference. The MDM hub is a read-only system, meaning changes are primarily applied to the original MD sources. The implementation of the *coexistence style* is similar to the consolidation style. It stores data in central MDM systems and updates it in its respective source systems. Compared to the consolidation style, it is a loosely coupled environment as MD can be created and updated in the central system and different systems and applications. Lastly, the centralized style is hugely invasive to the business and applications infrastructure. It is considered to be a centralized data source with a single source of truth. Yet, it provides great control and security with centralized governance.

Applying traditional MDM approaches disconnects the data perspective from the conceptual perspective. Traditional MDM stores the key-business objects of the conceptual level in a business glossary; the business object glossary is defined enterprise-wide or per business unit [41]. For the data level, traditional MDM approaches only link the data applications to each other and not to the conceptual level [40], [42], [43]. Consequently, lots of reports are generated with no consistent way for naming the data fields, which results in the need for high time and effort invested for the MD analysis by stakeholders. For example, the single managed MD hub of the consolidation style is a Read-only system that does not control all data centrally. Updates are locally executed on the MD sources [38]. Thus, traditional MDM is limited in satisfying **R1**.

According to [28], **R2** is an important prerequisite for MDM success. It is a major challenge to agree on definitions of key data items and to involve all stakeholders in MDM [28]. Traditional MDM systems, expert questionnaires, and interviews evaluate the satisfaction of the domain, and business experts [44]. This leads to increased effort as it is time-consuming to analyze and evaluate the questionnaires and interviews.

Despite SC stakeholders' efforts to understand, control and optimize the SC operations, disruptive events occur and rattle the expected SC performance. DMP, as part of the SCM, enables the mitigation of such events and the control of SC behavior before, during, and after.

2.1.3 Supply Chain Disruption Management

SC disruptions as described by [45] are events that modify the flow of goods and materials, hindering the SC's overall objective of producing and delivering services and goods to end-customers. In fact, [46] defines SC DMP as the process to discover the disruptive event, recover from the effect, and potentially redesign the system triggered by recovery learning outcomes. Namely, discovery refers to the point in time when SC stakeholders become aware of the disruption [47]. Then, disruption modeling of the system dynamics, e.g., via Petri nets, in simulation tools, is essential in order to analyze expected consequences and effects of the discovered event [48]. For instance, relying on the system dynamics simulation model implemented in the AnyLogic 8 tool [49] demonstrates the behavior of a multi-echelon SC responding to different end market scenarios [50].

Further, SC stakeholders choose the most effective recovery strategy to minimize the impacts of the disruption [47]. Thus, the recovery performance analysis evaluates the SC's ability to repair and return to the pre-disruption phase. Based on the evaluation's learning effects, SC stakeholders can rethink the SC design and operation processes and potentially decide on changes allowing for more resilience, e.g., increasing production capacity or applying multiple sourcing strategies. DMP entails the integration of highly heterogeneous data sources. For instance, [51] integrates data from bill of material, part routing, inventory levels, and plant volumes to map the SC and accordingly assess the impact of a disruption originating anywhere in product manufacturing and delivery. Also, [52] examines data from raw materials procurement along with inventory management systems to test the effect of various strategies in establishing resilience.

The degree of management needed for SCs depends on the complexity of the product, the number of available suppliers and customers at each level, the availability of raw materials, and the length of the SC [2]. In semiconductor markets, SCs are increasingly challenged by growing complexity as well as rising data volume and complicated structures.

2.2 Semiconductor Supply Chains

This section gives an overview of the semiconductor SC. First, we present the industry characteristics. Then, we explain the entailed bullwhip effect due to long leads and fluctuating demand. Finally, we show strategies to tame the bullwhip effect, e.g., dynamic pricing and forecasting.

2.2.1 Industry Characteristics

The electronics market is known for being volatile due to a large mixture of requirements and abrupt changes in demand [53]. The semiconductors amongst electronics products are also characterized by a short product life cycle, and rapidly changing technologies leading to high demand uncertainties [54]. Figure 2.3 shows Infineon's semiconductor E2E SC relying on the SCOR model with the characteristic processes: *Source, Make, Deliver, Plan,* and *Return* as described by [55]. The SC starts with the *Source* process from the suppliers' suppliers and ends with *Deliver* at the customers'

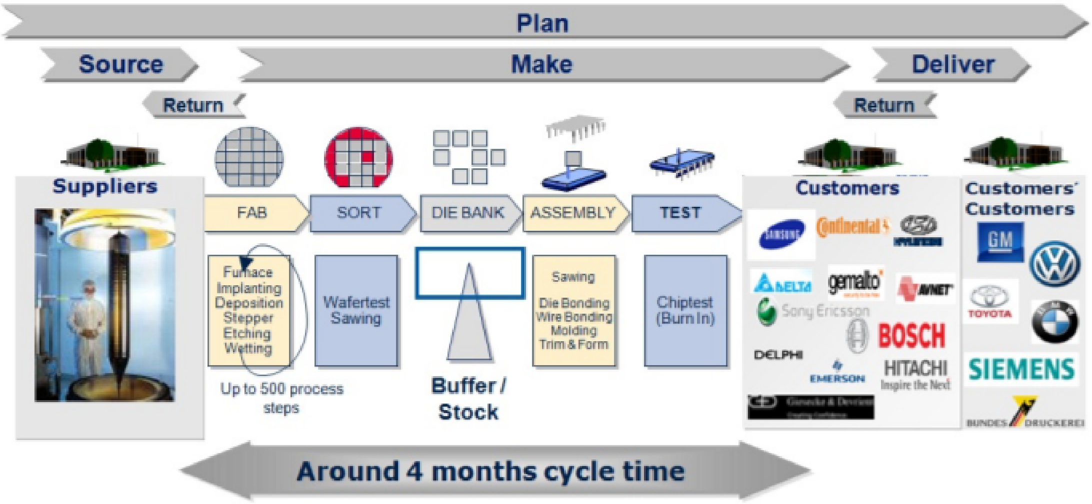


Figure 2.3: Infineon's end-to-end supply chain with SCOR: Source, Plan, Make, Deliver, and Return processes. The Make process incorporates the backend and frontend manufacturing details.

customers. Additionally, matching the demand and capacity requires effective planning and optimal utilization of production capacities, i.e., *Plan* process.

Manufacturing Process

Due to the complexity of the *Make* process in the semiconductor domain, the manufacturing lead-time can take up to 16 weeks. It is typically divided into two main stages: Front End and Back End. The Front End consists of the wafer fabrication and sorting, requiring up to twelve weeks. The wafer is built layer by layer, where each layer includes deposition, lithography, etching, ion implantation, and electroplating. The number of process steps varies depending on the product, as some of them are repeated several times at different stages of the process [56]. Afterward, the electrical die sorting (EDS) tests when the wafer fabrication process is complete and electronically sorts out any chips with quality defects. Only qualified chips are sent to the packaging plant.

The second major stage of the manufacturing process is the backend, where the wafers are diced into individual chips, which are then put into an appropriate package during Assembly. After Assembly, another testing stage ensures the quality of the final product (Testing) before distributing it to the customer. Package testing includes both quality control and reliability control. Quality control mainly involves detecting the availability of the chip after packaging, i.e., the chip's performance. Reliability testing consists of testing the parameters related to the reliability of the packaging, e.g., temperature and humidity test. Every step of the production involves different facilities spread around the world. For instance, a real-life example of the process of manufacturing a successful chip is Wafer fabrication in Germany, with certain process steps carried out in Malaysia, grinding in Austria, assembly in South Korea, and the final test in Singapore [57].

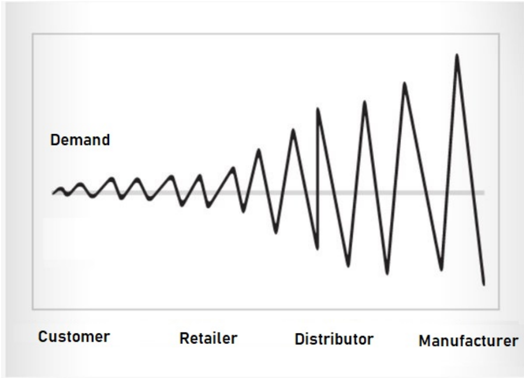


Figure 2.4: Bullwhip effect: increase of demand variability as the demand information is transmitted from customer to retailer to distributor to manufacturer to supplier.

Packing Information

After manufacturing the semiconductor product, it is packaged to be sent to the customer through the Distribution Center (DC). The packing info is a data object that includes information on how a product is stored and shipped to the customer. It contains a list of packing materials used for packing a product and additional packing information, like packing weight and outer box dimensions (width, length, and height). Having correct and complete values is important for customs declaration in deliveries to various countries to avoid triggering errors in the system during the DC automation activities.

The semiconductor industry is characterized by complex SC structures due to its wide range of customers with fluctuating demands for products, long production lead-times, and short product life cycles [58]. Therefore, distortions in forecasts and order management processes arise; this is known as the bullwhip effect [59].

2.2.2 Bullwhip Effect

The bullwhip effect is the increase of variability of orders as they move up the SC from retailers to wholesalers, distributors to manufacturers to suppliers [60], as shown in Figure 2.4 by [61]. Especially in the complex semiconductor SCs, the bullwhip effect is amplified. One of the main causes of the bullwhip phenomena is the local treatment of demand information. Each step of the SC anticipates the coming requests without correlating the end customer demand. [62] elaborates that the demand fluctuation caused by the bullwhip effect leads to the creation of excessive inventories to avoid shortages and revenue losses. Also, SC stakeholders tend to increase the production capacities and/or change the capacity plans in order to satisfy peak demands.

SC integration of planning and control enables the reduction of inventory and increases customer-service levels [63]. Centralized information and elaborate information integration among SC partners, as well as the flow of information in SCs, are crucial for carrying out effective exchanges between parties [64], thus limiting the increase in demand variability, i.e., reducing the bullwhip effect [65].

Apart from this, to tame the bullwhip effect, balance demand and supply, keep utilizing capacity in a profitable way, and guarantee customer satisfaction, companies resort to revenue management ideas such as dynamic pricing [59], i.e., exploiting faster delivery to generate revenue.

2.2.3 Dynamic Pricing: Lead-time-based Pricing (LTBP)

Dynamic Pricing (DP) describes the firm's practice to charge various customers different prices for the same products [66]. DP is defined as the tool maximizing the company's revenue and/or profit [67]. DP strategies subsume price maximization to create a balance of demand/supply. DP factors depend on the industry-specific demand factors, on the customer behavior, and characteristics. For instance, in the hotel industry, prices vary based on seasonal changes, local events, and location [68]. For the automotive industry, high inventory levels drive firms to decrease the prices to drive demand up, which reduces inventory holding cost [69].

In semiconductor manufacturing, lead-times are longer than customer order lead-times (the time interval between order entry and requested delivery date). Thus, fulfilling order requests that go beyond promised delivery time is costly [70]. Leadtime-based pricing (LTBP) establishes faster deliveries for higher prices to avoid high inventory capacity, tame demand variability, and compensate for the long manufacturing times [71]. [59] proposes an SC planning framework for revenue management that consists of solutions for demand steering and dynamic pricing where the pricing algorithm relies on the data from order lead-time measurement.

Lead-time Definitions

The *order lead-time* (OLT) at each stage of the order is the time frame between the entry of an order to the time point it passes a designated stage [71] as shown in Figure 2.5. A *requested order lead-time* $OLT_{Requested}$ is the difference between the *Customer Request Date* for an order, i.e., when the customer wishes to receive the order, and the *Order Date* (when the customer placed the order). A *confirmed order lead-time* $OLT_{Confirmed}$ is the difference between *Customer Delivery Date*, i.e., the confirmed order delivery date and the *Order Date*. *SDT* is defined as the difference between the *Standard Delivery Date* and the *Order Date*. The customer can expect the lead-time as the latest time the order will be delivered. In practice, customers request their orders earlier than *SDT*, i.e., $SDT > OLT_{Requested}$. Moreover, due to the highly-competitive semiconductor market, manufacturers usually sell their products with a shorter $OLT_{Confirmed}$, closer to $OLT_{Requested}$. The potential of lead-time-Based Pricing (LTBP) is seen in the difference between the *Standard Delivery Date* and the *Customer Delivery Date*. Namely, faster delivery, $SDT > OLT_{Confirmed}$, is exploited by offering LTBP as part of dynamic pricing.

LTBP models which identify the price premium based on faster delivery are still under development. In fact, some rely on simple mathematical functions, e.g., linear, concave, convex. For instance, [72] introduces a convex model and explains the rationale behind is that both opportunity costs for the manufacturer and for the customer increase with decreasing lead-time.

$$P_{Premium} = \alpha * \log(\frac{OLT_{Confirmed}}{SDT}) \tag{2.1}$$

The authors propose Equation 2.1 where α is a factor obtained from simulating customer order behavior for maximizing revenue for the company and $P_{Premium}$ is the price change or the added portion to the original price of the order. Other LTBP models attempt to tailor the dynamic price based on further customer portfolio artifacts, e.g. customer class or impact on the company's revenue. Applying dynamic pricing potentially generates conflicts with customers, affecting their satisfaction and harming the firm's long-term relationship with the customer and ultimately its success [73].

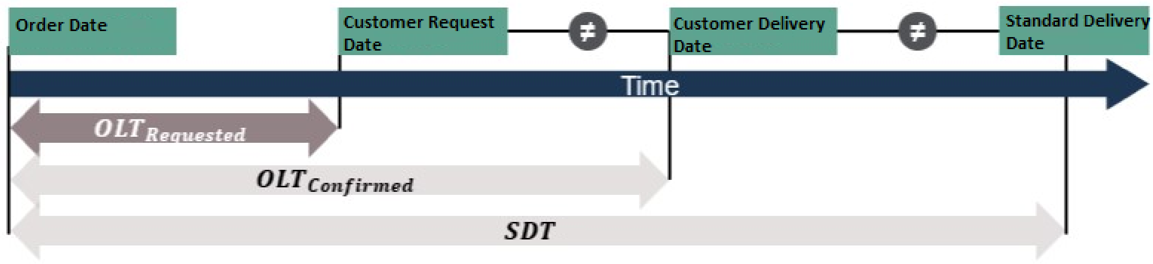


Figure 2.5: The definitions of lead-times with OLT as order lead-time and SDT as standard delivery time. The customer request date is sooner than the standard delivery date. The dynamic pricing potential relies when customer delivery date is shorter than standard delivery date.

2.2.4 Customer Relationship Management

Customer Relationship Management (CRM) refers to building one-to-one relationships with customers that can drive value for the firm [74]. Consequently, customization of requirements increases maintenance of customers [75]. Namely, by segmenting customers into portfolios, an organization can better understand the relative importance of each customer to the company's total profit [76]. Typically, CRM enables customer profiling and delimits characteristics for customers and how we can use these characteristics to determine the price as part of RM [77]. Customer Portfolio Management (CPM) is based on financial performance as well as strategic and economic criteria, e.g., customer account types or classes and regional importance. For each criterion, the company chooses measurable characteristics to segment customers accordingly. For example, customer classes distinguish customers according to their impact on business, based on the average revenue for the current and previous fiscal year, the volume of purchases, potential sales, the prestige of the account, and market leadership.

Especially in semiconductor manufacturing, a competitive domain, meeting customer-specific requirements requires maintaining close associations with customers to identify their specific needs. In fact, CPM affects new product development and product life cycle duration as it reflects customers' needs [78]. Because of the short product life cycle trait for this domain, it is especially important for manufacturers to act fast, understand customer order behavior and maintain good relations with the customer.

2.2.5 Customer Order Behavior

To be successful, semiconductor companies need to manage high demand volatility across the market. Disruptions and the bullwhip effect can also amplify demand changes and complicate demand planning. Since semiconductor production takes several months, a quick reaction to order and demand changes from customers and delivery on short notice is not feasible. Further parallelization to ramp-up semiconductor production is not possible due to semiconductors being built layer after layer. Also, building more manufacturing sites to react to market changes is capital intensive and takes several years and can therefore only be considered a long-term strategy. Having high inventories to deal with customer demand is not feasible since it is very costly for semiconductor manufacturers.

One common way to partially mitigate the previously-mentioned issues and enable better demand

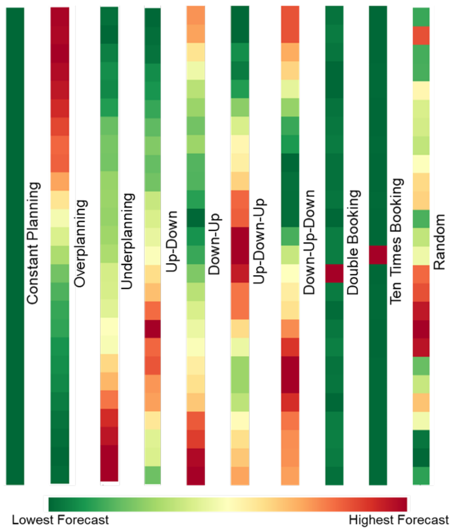


Figure 2.6: Customer order behavior patterns for a single delivery week. the green color shows the lowest forecast value for a customer while the red shows the highest.

planning is to allow customers to report their forecasts to semiconductor manufacturers. However, oftentimes the customers don't accurately estimate their future demand or present a tactical demand. A customer forecast that is too high or too low can hinder successful demand fulfillment. As one possible strategy, semiconductor manufacturers create their own forecasts for production based on customer demand to achieve demand fulfillment.

Behavior Pattern Prediction

Grasping Customer Order Behavior (COB) increases the data transparency and can help achieve more accurate forecasts. Previous work on COB with deep learning demonstrated the advantage of Convolutional Neural Networks (CNNs) and synthetic data generation for classifying order patterns [79] to improve understanding of customer behavior. [79] utilizes heat maps of the customers' order behavior that a CNN classifies according to predefined patterns. The authors analyze customer demand forecasts and forecast development over time.

Figure 2.6 presents exemplary patterns that have been identified in the heat maps, where each heat map represents a specific delivery week. The y-axis annotates the 26 weeks before the actual delivery week as communicated by the customers [79]. A dark green color represents a low forecast value, and a dark red color a high forecast value. Therefore, e.g., a heat map with an Overplanning pattern shows a customer who provides a high forecast long before the actual delivery and decreases the forecast closer to the delivery window. In total, ten patterns have been identified for single delivery weeks, as detailed by [79]. Those patterns are synthetically created without having noise included.

Context Information in Multi-stream Neural Networks

External context information (market situation, disruptions, or the customer's financial situation) indicates changes in the customer situation, which can lead to varied order behavior. Consequently, such context needs to be included for COB classification.

No.	The reference model requirement
R1	Model a base system: resources description and behavior
R2	Include products manufactured by the SC
R3	Represent customers responsible for order generation
R4	Model demand information that allows planning decisions
R5	Incorporate a simple planning and control system
R6	Show information flow for decision-making entities
R7	Include control flow to show how planning and control instructions are communicated
R8	Be easily understood and used, e.g., XML data structures are appropriate
R9	Incorporate performance measures

Table 2.3: Requirements for a reference model for semiconductor supply chains and abbreviations e.g., R1, R2.

Utilizing multi-stream neural networks to augment a neural network with context information is a common approach. One stream handles the main classification task, and the other one handles the context information. [80] uses context information to enhance classification of art with CNNs. Also, multi-stream neural networks are commonly used for action detection systems [81] [82] [83]. [81] introduces CNNs based on two streams for the recognition of human actions in videos. Two separate streams are used to handle spatial information, like objects and scenes, and temporal information, like the motion across frames.

2.2.6 Semiconductor Supply Chain Modeling

SC modeling is essential for semiconductors as it provides domain representation that allows to understand the complexity and test the behavior in empirical environments. [84] emphasizes the criticality of detailed modeling in semiconductor SC simulation. Authors elaborate that abstracted models can potentially lead to inaccurate determination of operational measures. [85] provide detailed separate simulation models for inventory management, capacity and production planning, demand planning and fulfilment.

Otherwise, [86] proposes to decompose the SCs systems into two or three layers. [87] divides the semiconductor SCs in four layers. Level one, the most granular, includes the interactions across materials and resources at the equipment level. Level two represents the manufacturing site with the major classes of work area, demand, lot and route. Level three depicts a broader view of the production network, including the internal supply chain (frontend, backend, distribution centers, and production partners) but without customers. Level four depicts the end-to-end supply chain.

Alternatively, [88] proposes a mathematical model for the semiconductor industry SC consisting of production and distribution chains. [89] relies on a combination of methods for control engineering in the semiconductor industry. The authors use a block diagram (to describe the overall concept of a complex system), Laplace transformation (to handle system inputs), and Transfer function (to represent the dynamic behaviour of production systems). [90] highlights that in the literature, there are no reference models for semiconductor SCs. The authors emphasize that reference models in this domain enable benchmarking of operational processes and planning algorithms. Thus, they list the requirements, shown in Table 2.3, for such a reference model, and propose a simulation reference model.

CHAPTER 3

Overview of Semantic Supply Chain Management

This chapter outlines the background knowledge for semantic SCM. First, we introduce the semantic technologies and the underlying standards. Second, we examine existing approaches and definitions for semantic SCM.

3.1 Semantic Technologies

We rely on Semantic Technologies in our implementations. In this section, we present the necessary terminology describing the pillar semantic artifacts: RDF, RDFS, OWL, SPARQL, and ontologies. Then, we elaborate on semantic data integration concepts.

3.1.1 Semantic Web Standards

The World Wide Web Consortium (W3C) refers to the SW in order to fulfill its vision of having a Web of linked data. The following standards, established by W3C for SW, enable users to publish, organize, and link data using vocabularies, queries, and reasoning. The SW stack is an illustration of the building blocks and standard technologies of the SW. The following sections describe the standards used in this thesis.

Resource Description Framework (RDF) is a standard model for data interchange on the web. The RDF relies on a directed, labeled graph to structure the content, where resources are graph nodes and edges are the relations between them. Nodes and edges are uniquely identified by the Universal Resource Identifier (URI). Nodes are referred to as classes, while edges/predicates as properties. URIs are bound to prefixes that enable conciseness, e.g., `http://www.w3.org/2000/01/rdf-schema#` is linked to the PREFIX *rdfs*. The RDF graph consists of a set of RDF triples where each triple contains a subject node, predicate edge, and object node.

Resource Description Framework Schema (RDFS) is a semantic extension of RDF that provides mechanisms for describing groups of related resources and the relationships between them. Figure 3.1 shows a simple example of an RDF graph with ':' as a PREFIX for <`http://www.example.org#`>.

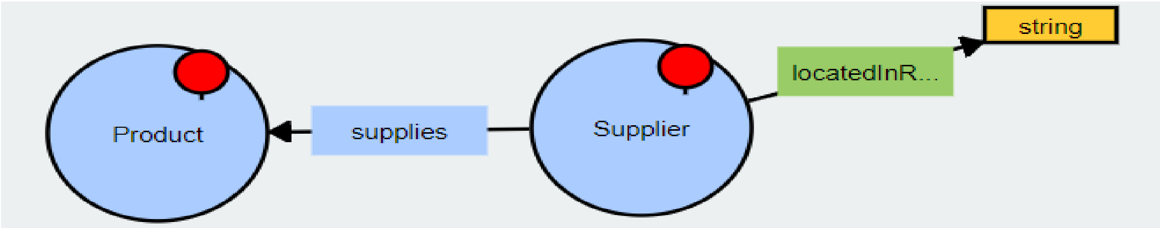


Figure 3.1: A graphical visualization example of the RDF Graph, where Supplier and Product classes are linked with supplies object property and islocatedInRegion describes the location of a Supplier.

:Supplier and *:Product* are instances of the class *rdfs:Class* and are linked to *:supplies*, an instance of *rdfs:Property*, with *rdfs:domain* and *rdfs:range*, respectively. Similarly, *:locatedInRegion* characterizes a *:Supplier* class with a region described by a string. Literals are used for values such as strings, numbers, and dates. Resources in an RDF graph can be described, using RDFS, as subclasses, thus enabling one to build hierarchical class definitions. For example, *:Supplier rdfs:subClassOf :ManufacturingSupplier*.

Web Ontology Language (OWL) builds on RDFS and provides a language to define ontologies, i.e., OWL documents. Ontologies model complex knowledge, by relying on triples. OWL supports different syntax to enable an exchange of ontologies among tools and applications. The most common is Turtle syntax as it offers more readability, as shown in the example Listing 3.1.

Listing 3.1: RDF Turtle syntax example for the visualization in Figure 3.1.

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>.
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
PREFIX xml: <http://www.w3.org/XML/1998/namespace>.
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>.
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
PREFIX : <http://www.example.org>.

<http://www.example.org> rdf:type owl:Ontology.
:supplies rdf:type owl:ObjectProperty;
          rdfs:domain :Supplier;
          rdfs:range :Product.
:locatedInRegion rdf:type owl:DatatypeProperty;
                 rdfs:domain :Supplier;
                 rdfs:range xsd:string.
:Supplier rdf:type owl:Class.
:Product  rdf:type owl:Class.
```

SPARQL is a query language for RDF, used to express queries across diverse data sources. A SPARQL query, as shown in Listing 3.2, consists of three parts, marked by the capitalized keywords: *PREFIX*, *SELECT*, and *WHERE*. Firstly, the *PREFIX* defines the namespaces and their abbreviations. Secondly, the keyword *SELECT* determines the output of the query. They are identifiers of query variables for which we want to get return values. SPARQL supports four types of selection queries:

SELECT, CONSTRUCT, ASK, and DESCRIBE. Lastly, the actual query is introduced by the *WHERE* keyword and is constructed by triples. Variables are marked with a question mark [91].

Listing 3.2: SPARQL query example to select supplier located in Europe and the supplied product.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX : <http://www.example.org#>
SELECT * WHERE {
  ?supplier :supplies  ?product.
  ?supplier :locatedInRegion  'Europe'.
}
```

3.1.2 Semantic Data Integration

In complex and interconnected domains, e.g., SCs where data is heterogeneous, dispersed, and big, semantic data integration becomes relevant [92]. Semantic data integration allows combining data from distinct and distributed sources using a data-centric architecture based on an RDF model. [93] introduces the term "information interoperability" to communicate and exchange information effectively as well as to integrate different information systems, applications, and services. The authors describe an ontology-based data integration concept. The use of ontologies aims at providing richer semantics and means to overcome semantic heterogeneity problems by deriving implicit knowledge at the schema level.

Ontology-Based Data Access (OBDA) is one approach that implements semantic data integration via three components: ontology, data sources, and mapping [94]. The ontology is given in terms of a schema representing the formal and conceptual view of the domain. It is a network of concepts, properties, and links describing a domain. The data sources exist in the underlying information systems. The mapping defines the relation between the data and the ontology. The output of the OBDA is a KG which contains ontologies and instance data. In semantic data integration, an ontology enables information from one resource to be mapped accurately at an extremely granular level to information from another source [95]. Thus, leveraging ontologies for semantic data integration help overcome dispersed data silos in SCs. In the following section, we present ontologies and vocabularies for SCs.

3.2 Semantic Supply Chain Management

Due to the diversity, dispersion, and complexity within an SCN, visibility, integration, and interoperability are challenging. SC visibility relates to the ability of the focal company, i.e., the SC leader, to access/share information related to the SC strategy and the operations of all SC partners [96]. Thus, SC visibility can improve strategic performance directly [97]. Semantic SC models rely on ontologies and KGs to ensure information exchange and to allow partners to reach visibility and agile information integration.

3.2.1 Semantic SC Models

SC semantic models support SC data integration by providing a comprehensive and explicit understanding of business-related concepts [98]. Semantic modeling allows SCs to overcome the siloed paradigm and to blend and consolidate data from dispersed data sources [99].

The literature review by [100] lists existing SC ontologies as an attempt to represent the complexity of the SC domain. The authors identify that a semantic model includes the strategic, tactical, and operational views of the SC **(C13)**. Moreover, an SC ontology covers an organizational extent, i.e., internal or external SC **(C14)**. The model incorporates an industry sector **(C15)**, has a purpose **(C16)**, and supports SC applications **(C17)**.

Related Work for Semantic SC Models

[99] developed Onto-SCM to provide shared terminologies for representing SC concepts and relations. Also, [101] formulated a comprehensive domain ontology to improve SC management efficiency by facilitating data integration. As identified by [100], all the existing SC ontologies cover the strategic level of granularity; none of the models support tactical and operational levels **(C13)**. Also, the authors explain the lack of inductive and collaborative modeling approaches **(C15)**. As well, the scope of SC, **(C14)**, is limited to the inter-business network. In an attempt to fulfill the shortcomings of existing semantic SC models, we study hybrid models that integrate SC aspects from various underlying base models.

Semantic Hybrid SC Models and Gap Analysis

We examine hybrid SC models that combine SCOR, E2E, and semantic SC models pair-wise. Table 3.1 lists the literature that studied SC hybrid models and identifies their gaps with respect to the concepts **(C1-17)**. The first column represents existing SCOR E2E models as shown in section 2.1.1. We highlight, in gray, the SC concepts that are not covered by existing SC independent models. In the gap analysis process, we consider different models as follows:

1. We examine models incorporating E2E **(C8-C12)** and semantic **(C13-17)** concepts. Long et al. [102] present a semantic model that subsumes SCN structure and covers multiple flows, develops and uses certain strategies, undergoes processes, uses multiple types of resources, and produces and uses several items. This work offers a semantic model addressing all aspects of an E2E SCN model **(C8-11)** except **(C12)**. Also, the authors include the tactical and operational granularity levels **(C13)**. Also, Suherman et al. [103] cover SC semantic model concepts **(C13), (C15), (C16), and (C17)**. Both proposed works cover an internal and external SC scope **(C14)**.

2. We analyze semantic SCOR models **(C1-7)** and **(C13-17)**. Zdravkovic et al. [104] describe the SCOR-Full ontology and its relations with relevant domain ontologies. Also, Petersen et al. [105] introduce the SCORVoc RDFS vocabulary to fully formalize the latest SCOR standard along with the key performance indicators (KPIs) defined by SCOR. Lu et al. [106] propose a product-centric SC ontology framework for facilitating the interoperation between all product applications involved in an extended SC. Fayez et al. [107] model an ontology for SC simulation modeling that enables the user to capture the necessary knowledge to build and generate simulation models. All models listed in Table 3.1 address SCOR SC aspects **(C4)** and

	SC-SCOR E2E-Network		Semantic SC-E2E-Network		Semantic SC-SCOR				
Model Covers:	Huan, Sheoran, and Wang, 2004	Xiao, Cai, and Zhang, 2009	Suherman and Simatupang, 2017	Long, Song, and Yang, 2019	Lin and Krogstie, 2010	Zdravkovic, Trajanovi c, and Panetto, 2011	Kirikova, Buchmann, and Costin, 2012	Lu et al.,2013	Petersen et al., 2016
(C1) Customer Interaction	No	No	No	No	No	No	Yes	No	No
(C2) Material Transaction	No	No	No	Yes	No	No	Yes	No	No
(C3) Market Interaction	No	No	No	No	No	No	No	No	No
(C4) Process Description	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes
(C5) SCOR Metrics	Yes	No	No	No	No	Yes	No	Yes	Yes
(C6) Management Practices	No	No	No	No	No	No	No	No	No
(C7) Software Products	No	No	No	No	No	No	No	No	No
(C8) Vertices	Yes	Yes	No	Yes	No	No	No	No	Yes
(C9) Edges	Yes	Yes	No	Yes	No	No	No	No	Yes
(C10) Various Material	No	Yes	No	Yes	No	No	No	No	No
(C11) Supply & Demand	Yes	Yes	Yes	Yes	No	No	No	No	No
(C12) SC vs SC	No	No	No	No	No	No	No	No	No
(C13) SC Granularity	Operational, Strategic	Operational	Operational	Tactical, Operational	Operational	Operational	Operational	Operational	Operational
(C14) SC Scope	E	E	I, E	I, E	I, E	I	I, E	I	I, E
(C15) Industry Domain	Generic	Generic	Generic	Generic	Generic	Generic	Generic	Generic	Generic
(C16) Model Purpose	Use network modeling to optimize SC performan-ce	Create an optimization model of cycle quality network	Examine technology enabler: cloud computing benefit SC	Provide guide of methodologies for complex SCN	Improve management of process via semantic interoperabil-ity	Overcome semantic inconsistenc-ies of the (SCOR) model	Compare the SCOR ontology to Value Reference Models	Contribute to enterprise semantic interoperat-ion	Facilitate information flows in networks for SC analysis
(C17) Model Application	Decision making in change managem-ent	Network for optim-ization environment protection	IoT applicat-ions	A four-echelon SCN: demonstra-te the application of a semantic model	Operation of three different business process models within logistics	Applicationin made-to-stock, made-to-order or engineered-to-order	SCOR ontology to model the information, and material flow	Make-to-Order process from body of grinding machine	Create synthetic benchmark to show the practicality of SCORVoc

Table 3.1: Gap analysis of existing SC models and described concepts, e.g., customer interaction and material transaction. E: External, I: Internal.

(C5). However, we note that **(C1), (C2), (C3), (C6),** and **(C7)** are not satisfied. The models include the operational granularity of an SC, **(C13)**. None of the models are industry-specific. However, they provide a purpose and an application: **(C15), (C16),** and **(C17)**.

Existing SCOR, E2E and semantic models alongside corresponding hybrid models are limited as they convey essential SC aspects in an isolated manner. In section 4.1, we present a semantic model that leverages ontologies, KGs and the SPARQL query language to provide an overall perspective of an E2E SCN, standardized SCOR processes and performance indicators.

3.2.2 Ontology-based Simulation Models

Simulation-based analyses are useful to map, benchmark, and improve SC operations [108]. However, the modeling and analysis of larger SCs proves to be computationally intensive and requires large amounts of data to output statistically valid results [85]. A common approach to mitigate the computational burden of DES-based simulation replications is to use meta-models. [109] proposes a meta-model-based Monte Carlo simulation to replace the DES model for production planning.

Research has also been undertaken to develop simulation object libraries, allowing rapid development of a reduced simulation model for SCs [110]. [90] proposes a simulation reference model to reduce the modeling and computational burden.

Moreover, recent testbeds related to semiconductor operations have been published both researchers and practitioners can use that to evaluate their respective approaches with a common playground while avoiding the modeling effort [111], [112], and [113]. Nevertheless, research has not resulted in a broadly reusable standard model accommodating a varied range of research questions. Ontologies enable the automation of the building process of a simulation model.

Related Work

[114] presents an Ontology-driven Simulation Modeling Framework (OSMF) that provides a visual programming interface to readily build, compose, and maintain distributed simulations. The key motivation is to facilitate simulation composability, integration, and interoperability. The OSMF concept is based on model libraries, comprising ontology and process templates with structural and behavioral information of reusable components, as well as reference libraries containing scalable domain models with reference process ontologies and reference information meta-models. Ontology libraries serve as well-structured, revisable knowledge databases.

Moreover, [115] develops CODES, a hierarchical framework to support component-based modeling and simulation. The basic idea of the framework is the component-connector paradigm, where connectors link the components allowing the exchange of data and messages. The framework allows the users to look for customized components, reuse the existing ones, and check the semantic and syntactic composition of the simulation system. CODES relies on an ontology called COSMO. The hierarchies of the ontology go in two main directions: since the ontology wants to be as general as possible and, at the same time, it wants to fit even the most domain-specific requirements, the ontology describes a set of components shared among all the domains and components specific to each application domain. Moreover, it also outlines the attributes and behavior of each component.

[116] discusses the development of Process Interaction (PI) Discrete Event Simulation (DES) ontologies named Process Interaction Modeling Ontology for Discrete Event Simulations (PIMODES) and Discrete Event Model Ontology (DeMO), see in Figure 3.2. Both ontologies were developed using OWL but with different approaches. PIMODES intends to support the interchange of simulation models as an ontology focusing on process interaction world view, while DeMO is developed as a DES Ontology focusing on DES world views.

DeMO's rationale considers that all discrete-event models have basic components, as well as mechanisms for how the models should run. Therefore, its structure begins with a base class, DeModel (discrete-event model). The sub-classes that follow are state-oriented, event-oriented, activity-oriented, and process-oriented models, which describe modeling formalisms. Subsequently, these particular formalisms serve as the base for a hierarchy of modeling techniques. Every subclass, representing a modeling technique in detail, has properties inherited from its base class, but with additional restrictions, and additional properties. To define a subclass of DeModel, one needs to relate the ModelMechanism subclasses to the ModelComponent subclasses.

Several concepts represented as subclasses of ModelComponent and ModelMechanism (such as state, event, time, etc.) are fundamental concepts in simulation and modeling. For instance, a process-oriented model has specific mechanisms, e.g., ProcessTriggering and ProcessEnabling mechanisms. This subclass subsumes different types of processes. Each process is modeled as a directed graph

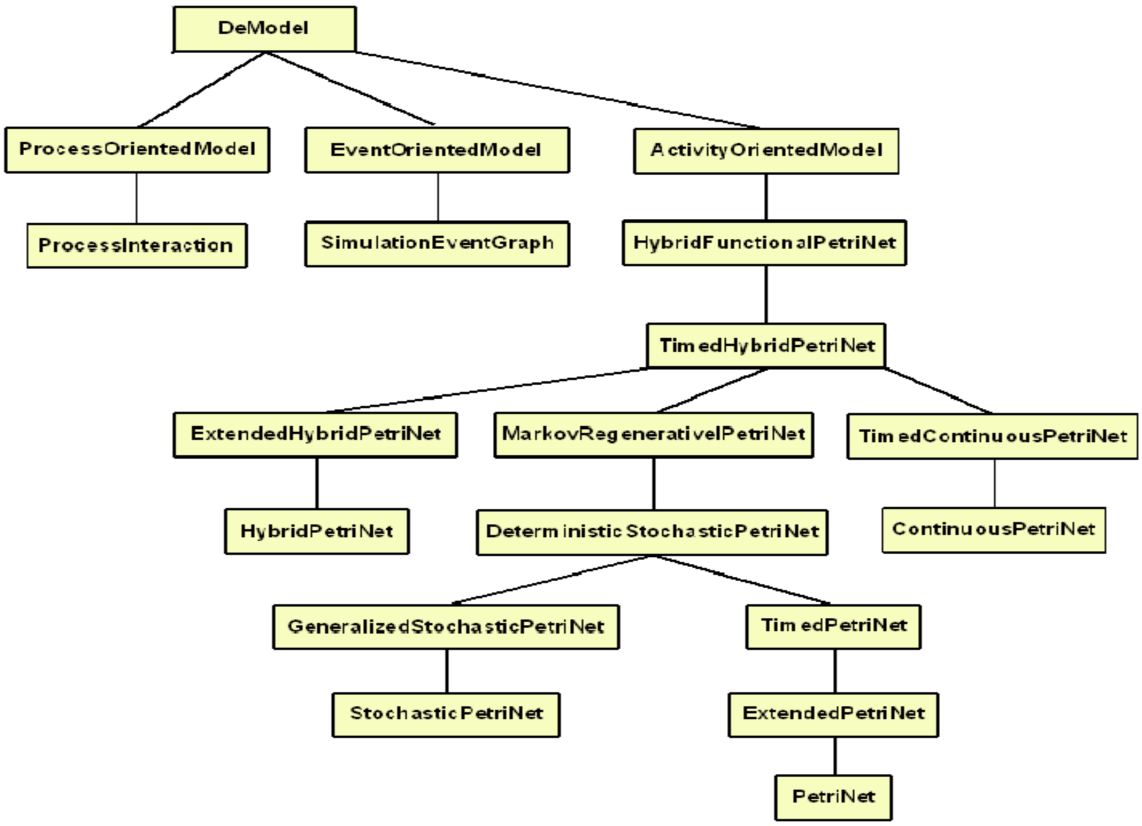


Figure 3.2: Extract from DeMO's taxonomy of Discrete Event Simulation (DES) modeling formalisms and techniques.

whose vertices are ProcessActivities, and edges, which link the ProcessActivities, are Transports. For example, *ProcessActivity has-Input-Transport Transport* and *ProcessActivity has-Output-Transport Transport*.

3.2.3 Semantic Semiconductor Models

The two main publications concerning the topic both focus on planning and control steps. On the one hand, [117] builds the ontology in order to manage planning and control phases in the virtual enterprise environment. The authors describe the requirements analysis and system specification for an order promising module. The core elements of an ontology for planning tasks in the context of semiconductor SCs are derived. On the other hand, [118] introduces an ontology to allow for an SC-wide interoperability of software agents that supports planning and control decisions in semiconductor SCs by means of a domain- and task-specific ontology. The authors rely on an ontology to propose a prototype named S^2CMAS, hierarchically organized, agent-based system that allows for decisions ranging from long-term capacity planning for the entire SCN to detailed scheduling decisions for single wafer fabrication facilities.

Semantic modeling provides a common language to standardize SCs. Semantic models for Master Data Management allow the creation of a golden record of data across an SC and synchronize different stakeholders' inputs [29], [30].

3.2.4 Semantic Master Data Management (MDM)

Semantic MDM generates an elaborate understanding of the domain. Semantic data integration for MD provides the means for stakeholders to make informed decisions regarding SC data in a methodical and efficient way.

[119] presents two polar implementations of possible approaches for semantic MDM, namely the *unified* and the *federated* approach. Both approaches semantically link the conceptual to the data model. The first creates a central conceptual model, i.e., Knowledge Base, which links the MD objects. The latter interlinks independent ontologies and instances without creating a central Knowledge Base. Applying the federated approach makes it possible to continuously ingest MD updates and new MD sources. However, it is harder to control data quality due to a decentralized Knowledge Base [119]. The authors prove that the presented approaches enable MD integration and link the conceptual level to the data level, i.e., **R1**. However, the proposed methods include the stakeholders' perspective but in a rather limited way.

According to [120], representing MD in a semantically rich way is an essential prerequisite for supporting different stakeholder viewpoints and enabling a multi-view MD analysis. [121] propose an ontology-based, multilevel product modeling framework that develops a generic product ontology that provides information to all stakeholders across the product life cycle by an ontology . The mentioned work focuses on the product MD only while it does not extend to Supplier and Material MD. [122] defines the core MDM ontologies as the logical MDM model containing core business entities and relationships. The authors include stakeholders' external perspectives, e.g., financial information for the domains affecting the SC. Existing semantic MDM approaches are limited as stakeholders' inputs, **R2**, are included in the preliminary phases of MD development and not only as feedback in further steps. To overcome the identified gap in existing semantic approaches, in section 5.1, we provide a knowledge-graph-based MDM approach that covers MD requirements **R1** and **R2**.

3.2.5 Semantic Disruption Management Process

The management and the evaluation of disruptions and their consequences on the SC require the integration of various distributed data sources, e.g., from manufacturing, order, and inventory management. [123] elaborates that SC digital twins enable integration to discover the link between SC disruption and performance deterioration. Namely, semantic models, one sort of digital twins, facilitate information exchange and allow SCs to reach full and agile information integration.

There exist several articles in the literature that devise semantic implementations to analyze SC performance during disruptions. [124] creates an ontology model to monitor and model risks, give early warning, and propose a procedure for assessing impacts on SC. Also, [125] presents an ontology-supported risk assessment approach for a resilient configuration of supply networks. Moreover, [126] provides an ontology-based decision support system to intensify the SC resilience during a disruption. Despite these developments, we note that existing approaches address DMP process steps in a rather isolated way, i.e., only one step of the process is incorporated, e.g., to model the disruption risk or to assess its impact. In section 5.2, we present a framework that leverages semantic data integration to incorporate all steps of DMP.

3.3 Semantic Web for Supply Chain Machine Learning Models

In this section, we provide an overview of the application of ontologies, SPARQL, and reasoning for the preprocessing phase of a ML algorithm. Then, we study KGs as input for a machine learning model.

3.3.1 Semantic Web Technologies in Preprocessing

A lack of data quality affects industries as it has an impact on daily activities like product deliveries or even specific projects like automation. A Machine Learning (ML) model and the preprocessing, the stage needed before model implementation, are able to give a solution to data quality problems. In the preprocessing stage, the main focus is the understanding and cleaning of the data to identify and eliminate inconsistent instances. This stage is crucial to provide quality data to the ML model. Then, the model is able to give accurate predictions to prevent the inclusion of incorrect values in a database.

The CRISP-DM Process Model [127] shown in Figure 3.3 provides an overview of the life cycle of a Data Mining (DM) process. The phases of data understanding and preparation represent the preprocessing stage. Here is where activities like data exploration and cleaning and feature transformation, selection, and normalization take place. The core goal of a preprocessing phase is the detection and removal of inconsistencies from the data to ensure sufficient data quality for the following analysis phases. However, while the preprocessing is crucial to ensure the good performance of the ML model, it is often referred to as “labor-intensive" and “time-consuming". Empirical methods for preprocessing require domain experts to fill the semantic gaps between the different DM stages [128]. Semantic Data Mining (SDM) uses an ontology, “an explicit specification of a conceptualization" [129], and exploits its formal semantics during the DM process [128].

Related Work

We summarize in Table 3.2 all the related work. [128, 130, 131] refer to the ontology as the source of prior knowledge in the form of constraints for the preprocessing stage. Additionally, the hierarchies and concept relations in an ontology assist in concept unification and in the detection of attribute interrelations [130]. [132] alludes to the use of integrity constraints while reasoning to benefit expensive processes such as data cleaning. These constraint violations, identified through the data validity checks of the logic reasoning [133], enable the detection of inconsistencies and outlier values. [134] addresses the difficulty of using an ontology for constraint checks due to the Open World Assumption. Hence, they propose the conversion of ontology axioms to SPAQRL Protocol and RDF Query Language (SPARQL) queries. Fürber and Hepp [135] propose a Data Quality Management (DQM) ontology for the formulation of standardized rules and a set of SPARQL queries for the classification of quality problems and generation of quality scores. These approaches intend to assess the data quality and automate the cleaning tasks. Kontokostas et al. [134] apply SPARQL templates for test queries to assess the quality of linked data resources. They determine the constructs applicable for constraint checking and select functionality, cardinality, class disjointness, domain, and range properties.

In another related contribution from Fürber and Hepp [136], they extend the use of SWT for DQM. In their work, they sustain that the use of SWT enables sharing the quality requirements between the different parties in an SC. Another contribution directed to improve the DQM outcome is from

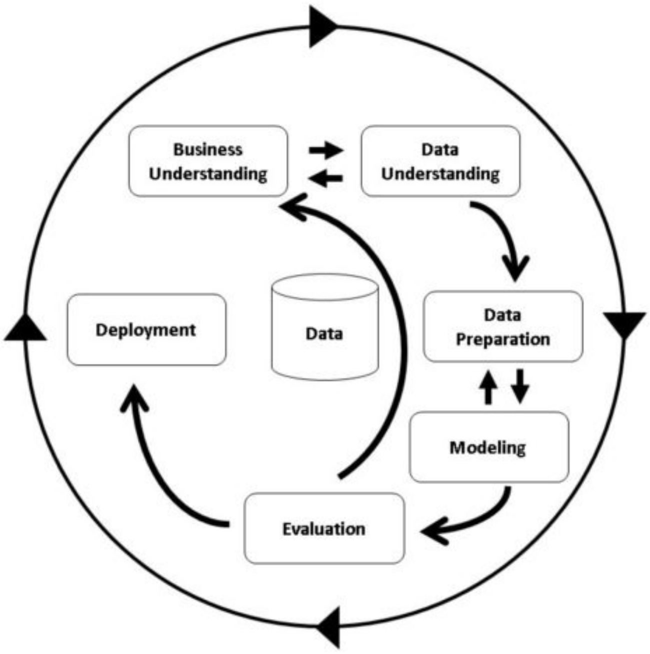


Figure 3.3: CRISP-DM: model of the life cycle of a Data Mining (DM) process: business understanding, data understanding and preparation, modeling, evaluation, and deployment.

<table>
<tr><th>Data Mining Phase</th><th colspan="2">Literature</th></tr>
<tr><td>Data Understanding</td><td colspan="2">[128, 130, 131]</td></tr>
<tr><td rowspan="2">Data Preparation: Inconsistency Detection</td><td>Ontology</td><td>[136] [137]</td></tr>
<tr><td>Reasoning</td><td>[132] [133]</td></tr>
<tr><td>Data Preparation: Quality Assessment and Classification using SPARQL</td><td colspan="2">[135] [134]</td></tr>
<tr><td>Data Preparation: Correction</td><td colspan="2">[138][139]</td></tr>
</table>

Table 3.2: Summary of related work for Semantic Web in preprocessing of machine learning models.

Brüggemann and Grüning [137], who use a domain ontology and consistency checking for duplicate detection and metadata annotations exploitation.

Once detected, the correction and data quality improvement are enabled. The preprocessing ontologies OntoClean [138] and OntoDataClean [139] demonstrate a task ontology that is triggered by a user query to select the proper method for data cleaning or data transformation.

In a more recent contribution, [131] introduces an ontology-based framework for detecting outliers and analyzing their potential causes. This framework assists all the phases for data selection, preprocessing, and transformation, up to demonstrating the direct impact of ontologies on the performance of a ML algorithm.

Research Gap

The analysis of existing work showed that the combination of an ontology approach and queries is commonly used for stipulating axioms, exploiting the reasoning, and setting up constraints in a preprocessing stage. While there are approaches focusing only on the detection of data insufficiencies ([128]-[137]), others already attempted automatic corrections [138], [139]. Nevertheless, to the extent of our knowledge, except for [131], the related approaches focus only on the preprocessing stage with the ultimate goal of improving the data quality. Only Gonzalez [131] incorporates semantic knowledge in the whole DM process and depicts its influence on the performance of a ML model. In section 6.3, we explore the inclusion of SWT in ML in the data cleaning and data understanding phases of the preprocessing stage and study the impact on the ML model performance.

3.3.2 Knowledge-Graphs As Input for Machine Learning Models

KGs are semantic models that capture the domain knowledge in a comprehensive way. Relying on KGs as inputs for ML models allows better understanding of the domain, thus, potentially improve the ML performance. Neural networks, such as CNNs, require dense numerical representations as an input. Thus, a KG with its entities needs to be embedded while still retaining semantic information, relationships, and literals, like numerical information. KG embeddings embed entities of the graph into low dimensional feature space and aim to preserve the structure of the original graph [140].

Embeddings allow the generation of vector representations for the KG associated with the subsequent downstream machine learning in a multi-stream network. One of the simplest embeddings is TransE [141], a translational embedding based on geometrical translations in the embedding space. Tensor factorization models such as ComplEx [142] apply tensor decomposition methods to derive tensors that capture features from the original tensors [143]. These can capture more complex asymmetric relations from the KGs than TransE. The mentioned models preserve the semantics and structure of the KG but ignore any literal information, like numerical information or text. Multimodal embeddings like KBLRN [144] and LiteralE [145] extend previous models with the ability to include literal KG information into the embeddings. In section 6.2, we present a methodology to use KGs, and corresponding embeddings to feed in a two streamline CNN.

CHAPTER 4

Semantic Models for Supply Chain Standardization and Benchmarking

The content of this chapter is based on the publications [146], [147], and [148] to answer the following research question

> RQ1: How can semantic models be used to standardize and benchmark supply chains?

The first section presents SENS a semantic SC model that integrates SC concepts and ensures SC topological and operational standardization. Second, we propose the Digital Reference, a reference vocabulary for the semiconductor SCs that not only serves as a standard for the domain but also eases the creation of other SC models, e.g., simulation models. We define a methodology that bridges ontologies to simulation models in order to enhance performance analysis.

4.1 SENS: Semantic Synthetic Benchmarking Model for Integrated Supply Chain Simulation and Analysis

Integrated modeling of the SC enables enriched behavioral analysis as it incorporates various SC concepts, e.g., topology, materials, metrics, and processes, in order to simultaneously give a perspective of SC structure, operations, and partners. There exist several SC models e.g., E2E SCN explained by [149] and SCOR [10] that include core SC concepts. For instance, the E2E SCN model provides an overall perspective of the SC partners as well as the flow of products, services, and materials which conveys SC structural coherence and resilience. The SCOR model provides common definitions of operational processes and metrics to enable SC standardization and benchmarking. As shown by the gap analysis in section 3.2.1, existing models tackle core SC aspects but still in an isolated manner, hence, limiting integrated SC behavioral analysis. Furthermore, the scarcity of integrated empirical data from SC members limits the study of the overall behavior. Firms do not disclose their connections to keep a competitive advantage or simply because there are not enough associated incentives or rewards [150]. Also, logs or data from one company are not enough to validate the E2E SC models.

We propose SENS, a standard SC model covering core aspects in an integrated fashion and tackling the shortcomings caused by isolated models. SENS is a semantic model that leverages ontologies, KGs and the SPARQL query language to provide an overall perspective of an E2E SCN, standardized

SCOR processes and performance indicators. SENS comprises SC partners and the relations between them representing the flow of materials and goods. Moreover, based on the production and inventory capacity model included, we provide a SPARQL-based demand fulfillment algorithm that mimics how an SC operates to achieve its ultimate goal of meeting end-customers' order requests. Additionally, we propose SENS-GEN, a highly configurable synthetic data generator that, based on input parameters, produces an exemplary instance of an SCN. SENS-GEN enables to generate SC data for various industries, e.g., automotive, dairy, determined by the topology and properties of the instantiated output KG.

4.1.1 SENS: Integrated Semantic Supply Chain Model

We describe SENS, an integrated semantic SC model to incorporate an end-to-end perspective of the SC including standardized SCOR processes and metrics.

SENS Ontology Model

The core of SENS Ontology depicted in Figure 4.1 is nodes representing SC partners. We model each partner as instance of the class *Node*, i.e., *Supplier*, *Customer* or *OEM*. SC nodes are organized in tiers, so we model this information using RDF triples of the form *Node belongsToTier Tier*. Accordingly, we distinguish between *SupplierTier* and *CustomerTier*.

The supply side is organized so that the raw material suppliers belong to the highest supplier tier, which is the most upstream tier, i.e., *SupplierTierN* [16]. Supplier nodes in low tiers are connected to suppliers in upstream tiers using the property *hasUpStreamNode* while on the customer side, end customers belong to the most downstream tier, i.e., *CustomerTierN*. Similarly, customer nodes in the low customer tier are connected to customers at downstream tiers with the property *hasDownStreamNode*. The links between nodes model contractual relations, organizing the flow of demand, materials, and products between SC partners. Likewise, *SupplierTier*s are connected with *hasUpStreamTier* while *CustomerTier* with *hasDownStreamTier*.

The Original Equipment Manufacturer (OEM) is the focal node responsible for assembling the product or getting it ready for distribution by delivering it to a warehouse or a wholesaler, followed by various distribution centers to the end-customer. The OEM is directly linked to the suppliers in *SupplierTier1* via the property *hasOEM* and *CustomerTier1* via *OEMhasNode*

Also, we model node operations with RDF triple statements of the form *Node hasProcess Process*, and the class *Process* has as subclasses the SCOR processes: *Source, Plan, Make, Deliver, Enable and Return*. Consequently, for each node, we model the SCOR KPI *hasResponsiveness, hasReliability, hasCost, hasAgility, hasAssetManagementEfficiency* to evaluate the operational behavior of this node based on the SCOR metrics standard. Furthermore, each node is described by data properties that either depict its performance, e.g., *hasCO2Balance* or its characteristics, e.g., *hasLocation*. We resolve node locations using geo-coordinates represented with the properties *hasLongitude, hasLatitude*.

Supply Chain Demand Fulfillment

The goal of an SCN is to fulfill end-customers' demand relying on production and inventory capacities. SENS models supply and demand and a SPARQL-based demand fulfillment algorithm to simulate SC production planning and scheduling.

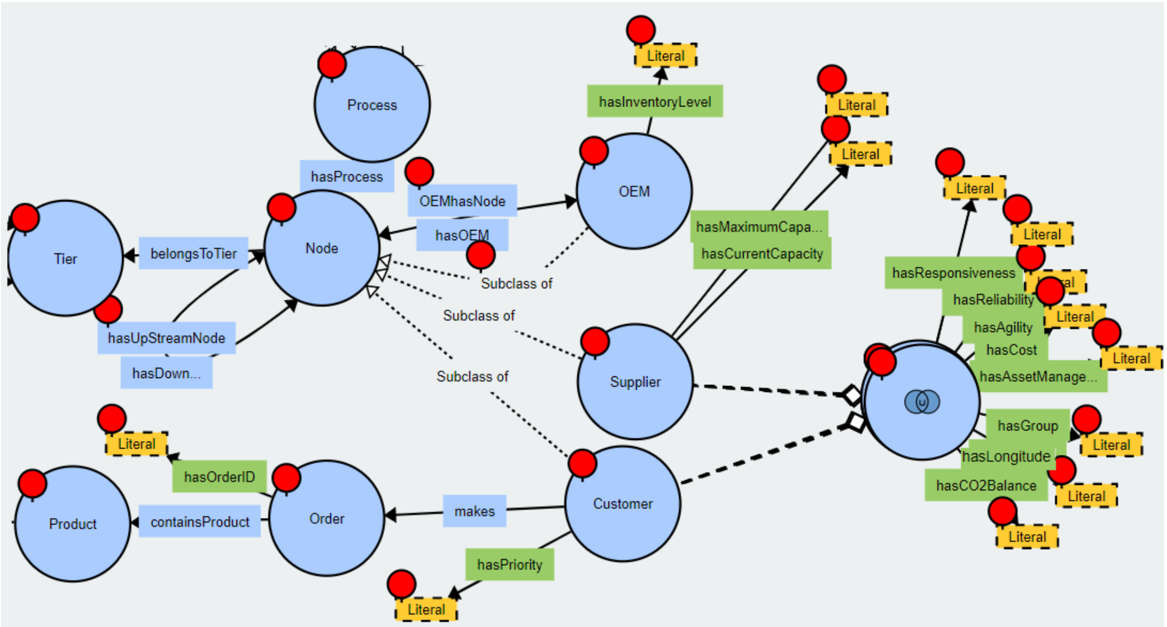


Figure 4.1: Depiction of the core concepts of the SENS ontology modeling End-to-End and SCOR supply chain concepts.

Supply Chain Demand. We model the demand as orders of products via triples of the following form: *Node makes Order*, *Order hasProduct Product*, *Order hasDeliveryTime xsd:dateTime* and *Order hasQuantity xsd:integer*. Moreover, customer orders are fulfilled depending on their priority modeled by *Node hasPriority xsd:integer*. Customer relationship management determines a customer's priority based on various factors, e.g., customer revenue, and contract type.

Supply Chain Capacity and Production. SC nodes produce and stock products in order to fulfill the demand. We rely on RDF-star, a framework to model in a compact way statements about statement [151]. RFD-star is widely implemented by tools such as GraphDB and Virtuoso; reification [152] is a viable alternative. The following list of triples models capacity and production of nodes in the SCN:

- *Node manufactures Product*: defines what products are manufactured by this node, e.g., *OEM manufactures Car*. *«Product needsProduct Product» needsQuantity xsd:integer* models the intermediate products needed to manufacture the final product. For instance, *«Car needsProduct Wheel» needsQuantity '4'* and *«Wheel needsProduct Rubber» needsQuantity 10m.*

- *Node hasTransportMode xsd:string*: SC nodes rely on one or more shipment modes, e.g., air cargo, maritime to transport products.

- *Node hasGroup xsd:integer*: in order to reduce purchasing prices and benefit from the supreme performance, suppliers capable of supplying the same products, i.e., belong to the same group, are exchangeable [153].

- *Node hasCapacity Capacity*: defines the availability of labour and resources to make a product by a node. The capacity is detailed by *Capacity hasProduct Product*, *Capacity hasCost xsd:integer*, *Capacity hasQuantity xsd:integer* and *Capacity hasTimeStamp xsd:dateTime*.
- *Node hasSaturation xsd:integer*: is the bottleneck defining the maximum capacity to manufacture at any time.
- *Node hasInventory Inventory*: models the node keeping stock of products describing the inventory using triples of the following form: *Inventory hasProduct Product; hasCost xsd:integer; hasQuantity xsd:integer; hasTimeStamp xsd:dateTime*.
- *Node hasDeliveryTime xsd:integer* indicates the time for a node to deliver to the customer after finishing production [154].

Demand Fulfillment. SCs follow a customer order-based strategy to determine its production scheduling [155]. We present a SPARQL-based demand fulfillment algorithm relying on backward scheduling, i.e., starting from the delivery time of an order and planning backward for its fulfillment. The **input** is incoming orders containing a standard product with constant repetitive demand. The **output** of this algorithm is a supply plan specific for each order modeled by *Order hasSupplyPlan SupplyPlan*. This plan is a scheduled capacity allocation for products among production facilities as well as the needed parts among suppliers, as shown by the following triple representation: *«SupplyPlan needsNode Node» getsProduct Product; hasTimeStamp xsd:dateTime; hasQuantity xsd:integer; hasUnitPrice xsd:double*. We determine the following base assumptions about the model:

- Nodes have a standard delivery time. When the node capacity is lower than the saturation limit, i.e., the node is operating far from the bottleneck, orders are fulfilled and delivered in constant time [156].
- The supplier selection process is based on respective capacities while suppliers' choice considers other factors, e.g., price, quality of service, CO2 balance [157].
- The demand fulfillment is a recursive cascading problem, e.g., nodes in *TierN* receive orders from nodes in *TierN+1*. Then the fulfillment either relies on the available inventory or production capacities. On the supply side, nodes in *TierN* decompose the product to needed intermediate products supplied by nodes in *TierN-1*. On the customer side, the same finished ordered products flow between nodes.
- SC planners determine the frequency of execution of the demand fulfillment algorithm.

In this sense, we consider the relationships between three tiers of the SC (SupplierTier1, OEM, CustomerTier1). The incoming demand to the OEM is the orders by customers in CustomerTier and is the aggregation of the incoming demand flow starting from the end-customer.

The following steps, executed at time t, outline the demand fulfillment algorithm. For conciseness, we show exemplary queries while we provide the detailed code and SPARQL queries in our accompanying technical report and GitHub repository in Appendix B and [8].

1. Listing 4.1: At t: Get orders by customer priority from CustomerTier1 where *O rdf:type Order*, *O hasProduct P*, *O hasDeliveryTime* $DT(O)$. The OEM has delivery time modeled by *OEM hasDeliveryTime* $LT(O)$ where $DT(O) - LT(O) = t$

Listing 4.1: Get orders by customer priority.

```
PREFIX : <http://www.example.org/SENS#>.
SELECT * WHERE {
  ?order :hasDeliveryTime ?dt.
  ?order :hasQuantity ?q.
  ?order :hasProduct ?p.
  ?custm :makes ?o.
  ?custm :hasPriority?prio.
  ?oem   :hasDeliveryTime ?lt.
FILTER (?dt-lt=t)
} ORDER BY DESC ?prio
```

2. If OEM inventory at *t hasQuantity Q(I)* suffices to fulfill the order quantity i.e., *O hasQuantity* $Q(O)$ and $Q(I) >= Q(O)$, then the order is fulfilled, a supply plan generated and the OEM inventory updated: $Q(I) = Q(I) - Q(O)$. Otherwise, we proceed with production in step 3.

3. Place a production order for the remaining $Q(I) - Q(O)$, if the OEM capacity at t is smaller than its saturation.

   a) Listing 4.2: Get all intermediate products and quantities to manufacturer P.

   Listing 4.2: Get all intermediate products for Product P.

   ```
   PREFIX : <http://www.example.org/SENS#>.
   SELECT * WHERE {
   << :P :needsProduct ?comp >> :needsQuantity ?quant.
   }
   ```

   b) Listing 4.3: Choose a supplier in SupplierTier1 with capacity for intermediate products smaller than their bottleneck at t_0 with $t_0 = t - LT(S)$, where *Supplier hasDeliveryTime* $LT(S)$. This means that the supplier has the capacity to produce the intermediate products at t_0 to reach the OEM at t to manufacture and fulfill the order at its delivery time $DT(O)$. Only if suppliers are chosen for all intermediate products, then the order is fulfilled and a supply plan generated. Otherwise, the order is not fulfilled.

   Listing 4.3: Get Supplier capacity for intermediate product at time t_0.

   ```
   PREFIX : <http://www.example.org/SENS#>.
   SELECT * WHERE {
     ?sup :hasOEM OEM1.
     ?sup :hasCapacity ?cap.
     ?cap :hasProduct ?p.
     ?cap :hasQuantity ?q.
     ?cap :hasTimeStamp ?t0.
     ?sup :hasSaturation ?sat.
     ?sup :hasDeliveryTime ?lt.
   FILTER  (?sat>= ?q +  tofullfil) &&  (t - ?lt= ?t0).
   }
   ```

4.1.2 SENS-GEN: Synthetic Supply Chain Knowledge-Graph Generator

We present SENS-GEN, a highly configurable data generator that relies on the SENS model to create a specific synthetic instance of an E2E SCN, incorporating SC concepts in an integrated manner. SENS-GEN allows examining and comparing the overall performance of SCs under various structural and behavioral changes.

SENS-GEN Parametrization

SENS-GEN receives input parameters to instantiate SENS ontology, i.e., SENS KG, that determines the topology and the performance of the SCN. Namely, the topology depends on the industry sector as it signifies the complexity of the products (the steps needed to manufacture), the variability, and the number of customers and suppliers. In fact, the topology is defined by the *Supplier_Tier, Node_Supplier_Tier, Customer_Tier, Node_Customer_Tier* parameters in Table 4.1.

The KG describes the behavior of the SCN through the values assigned to the nodes' data properties, e.g., *hasReliability, hasCO2Balance*. Namely, the capacity and inventory of the nodes allow the simulation of the demand fulfillment and evaluate the performance of this particular SC realization. The parameters assigned per node can be randomly generated from the range of values given, e.g., [1-5], or manually defined per node as an input. For conciseness, we show only the supplier side generation in Algorithm 1 (cf. the technical report [8] for the detailed code).

Algorithm 1 SENS knowledge-Graph generation algorithm

for $(n = 1;\ n <= Supplier\_Tier;\ n++)$ **do** ▹ Create tiers and nodes
    Create SupplierTier*(n)*
    for $(m = 1;\ m <= Node\_Supplier\_Tier[n];\ m++)$ **do**
        Create SupplierNode*(m.n)*
        Add SupplierNode*(m.n)*, *:hasGroup*, Random(1, Supplier_Group_Tier[n])
        for *Property P of SupplierNode(m.n)* **do** ▹ e.g., *:hasCO2Balance*
            Add SupplierNode*(m.n)*, p , Random(min_val, max_val)
            Generate saturation capacity, initial capacity and inventory

Generated Showcase Examples

We present two examples of SCNs from the automotive and dairy industries. Table 4.1 shows the parametrization of the model and the variation of topology and properties based on the industry. In Figure 4.2, we provide an example of a SCN in the automotive industry. We choose three supplier tiers, i.e., raw material, component, and system suppliers.The dairy SCN example in Figure 4.3 consists of one supplier tier, i.e., the dairy farms that are directly linked to the OEM. At the OEM, products are processed and packaged to be sent to retailers `CustomerTier1` then end-customers `CustomerTier2`, e.g., homes, restaurants.

There exist multiple KPIs to assess SC behavior, yet we focus on the SCOR KPIs as they enable a standardized performance evaluation and benchmarking. We set for the SCOR KPI, a range of [0-100]% as explained by [158]. The CO2 balance varies according to policies of countries where nodes are located as well as OEM environmental strategies but ranges between 30-45 Teragram (Tg)

Parameter *Triple Representation*	**Explanation**	**Automotive**	**Dairy**
Supplier_Tier	**SC** depth, manufacturing steps	3	1
Customer_Tier	**SC** distribution and sales interactions (OEM to end customer)	3	2
Node_Supplier_Tier	**SC** width, the suppliers providing materials for manufacturing	<2, 3, 5>	<3>
Node_Customer_Tier	**SC** customer availability	<2, 2, 4>	<2, 3>
Supplier_Group_Tier *Supplier hasGroup xsd:integer*	**Supplier** exchangeability to provide same products per tier	<1, 2, 4>	<1>
Node_Priority range *Node hasPriority xsd:integer*	**Customer** relationship management to prioritize customers	[1-3]	[1-3]
Node_Capacity_Saturation *Node hasSaturation xsd:integer*	**Node** maximum capacity to manufacture	[1-3] million unit	[0.5-1] million unit
Node_Delivery_Time *Node hasDeliveryTime xsd:integer*	**Node** time to deliver from node to node in following tier	[1-7] days	[1-3] days
Node_Initial_Inventory Node hasInventory Inventory	**Node** inventory at t=0	[10-50] thousand unit	[5-10] thousand unit
Node_Initial_Capacity *Node hasCapacity Capacity*	**Node** capacity at t=0	1 thousand unit	1 thousand unit
Data Property range *Node (hasResponsiveness, hasReliability, hasCost, hasAgilty, hasAssetMangmentEfficeny) xsd:integer*	**SCOR** KPIs. [158] explain how to calculate level 1 SCOR KPI from lower level metrics for SCOR processes	[0-100] %	[0-100] %
Data Property range *Node hasCO2Balance xsd:integer*	**SC** environmental performance	[30-45] Tg	[30-45] Tg
Data Property range *Node hasLongitude xsd:integer* *Node hasLatitude xsd:integer*	**SC** globalization (geographically dispersed network of nodes)	Long/Lat: [0-180/ 0-90]	Long/Lat: [90-180/ 45-90]
Customer_Demand_Frequency *Customer makes Order*	**SC** constant demand frequency	2	10
Product type and quantity per order *Order hasProduct Product* *Order hasQuantity xsd:integer*	**SC** orders variability and size	1: 100 thousand unit	1: 5000

Table 4.1: SENS-GEN parametrization and exemplary parameters for automotive and dairy industry.

[159]. The dairy SC is not as dispersed as the automotive industry as the products are easily perishable. Therefore, the range for longitude, latitude, and inventory is smaller, and the delivery time is shorter than in the automotive industry. However, customer orders are more frequent in the dairy industry but smaller product quantities.

4.1.3 Evaluation

We perform a two-fold evaluation. First, we prove that SENS is a semantic SC model that integrates core aspects of SC and deals with shortcomings caused by isolated models. Then, we provide an empirical performance analysis of the generated automotive SCN example introduced in section 4.1.2 and show behavioral changes under experimental conditions.

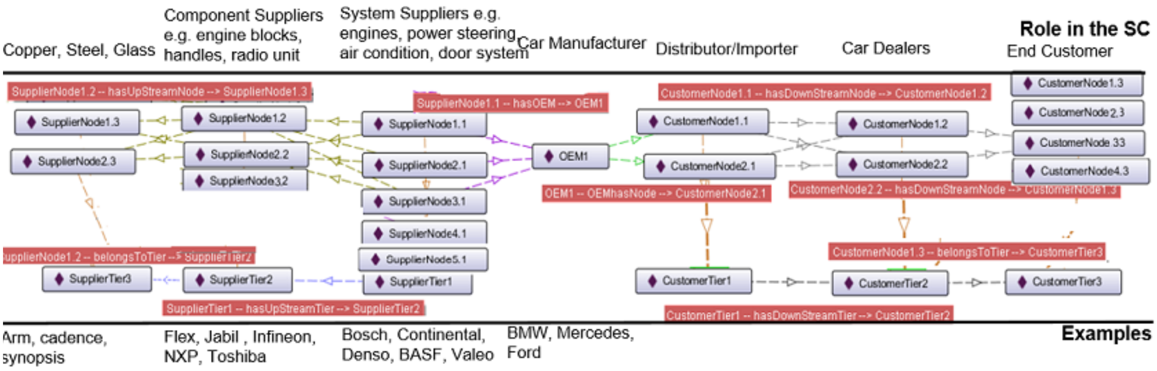


Figure 4.2: Automotive industry SENS knowledge-Graph example with three supplier tiers raw material, component and system suppliers as well as three customer tiers for distributors, car dealers and end customers.

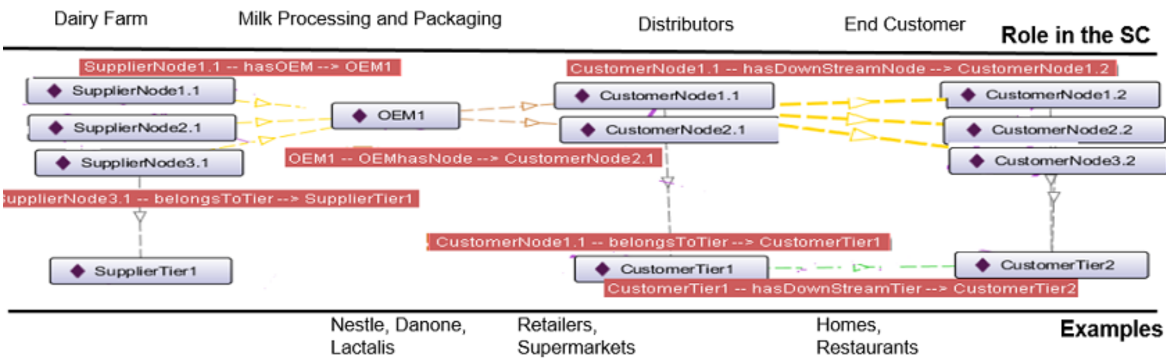


Figure 4.3: Dairy industry SENS knowledge-Graph example with one supplier tier, i.e., the dairy farms and two customer ties for distributors and end-customers.

SENS Model Validation

We validate that SENS is an integrated model by analyzing SENS coverage of SC concepts **(C1-17)** incorporated by SCOR, E2E and semantic SC models, listed in our literature assessment. In Table 4.2, we show the executed SPARQL queries and sample results from the automotive SENS KG. We note that the proposed SENS ontology and KG enable us to model and retrieve SC aspects **(C1-17)** except **(C6, C7)**. However, existing research in the domain implies that management practices and software products are hard to assess and thus not commonly represented in SC models. We can conclude that SENS integrates SC aspects covered by SCOR, E2E and semantic SC model.

SENS Knowledge-Graph Behavior Analysis

This section shows the benchmarking and integrated analysis in experimental contexts enabled by SENS.

Setup. We use the automotive SENS KG in Figure 4.2 created based on the parameters in Table 4.1. We run the demand fulfillment algorithm for 178 t (days), i.e. half a year.

	SPARQL Query: SELECT * WHERE	**Example Output Triples**
(C1) Customer Interaction	?customer :makes ?order. ?customer :hasDownStream ?c	Node3.2 makes OrderJZHu5 Node3.2 hasDownStream Node3.3
(C2) Material Transaction / (C10) Various Materials	«Product needsProduct ?p» needsQuantity ?q	«ProductA needsProduct Product1» needsQuantity 1
(C4) Process Description	?node :hasProcess ?process.	:Node3.2 :hasProcess :ProcessA. :ProcessA rdf:type :Make
(C5) SCOR Metrics	?node :hasResponsiveness ?r.	:Node3.2 :hasResponsiveness '24'
(C8)Vertices / (C9) Edges	?node a :Node ?node ?prop ?node2.	Node3.2 rdf:type Node Node3.2 hasDownStreamNode Node3.3
(C3) Market Interaction / (C11) Supply and Demand	Algorithm described in Section 4.1.1 detailed by [8]	
(C12) SC vs SC	Supplier **exchangeability** is modeled by *Supplier hasGroup xsd:integer*. Nodes share and compete over suppliers and customers.	
(C13) SC Granularity	**Operational**: SENS-SC spans SCOR operational processes e.g. *Source, Plan* and the supply plans address operational planning. **Tactical, Strategic**: Describing the performance via data properties e.g. *hasCO2Balance* enable analysis on different aggregation levels.	
(C14) SC Scope	SENS-SC models **Internal** node processes and **External** interactions by modeling the flow of supply and demand.	
(C15) Industry Domain	Model **parametrization** in subsection 4.1.2 to tailor the KG to any industry.	
(C16) Model Purpose	Provide a topology of SCN with detailed and **standardized** operational SCOR processes and relying on semantics for **interoperability**.	
(C17) Model Application	SC **behavior analysis** in empirical scenarios as shown in the following section.	

Table 4.2: SENS as an integrated semantic model covering SC core aspects.

Metrics. The following metrics are a sample of the SPARQL-based performance indicators to benchmark the performance of a semantic E2E SCOR SC. **Order Fulfillment**: Listing 4.4 evaluates how many orders the SC fulfills and generates corresponding supply plans. This metric evaluates the SC ability to achieve its goal of satisfying end customers' demand. Also, operating close to the saturation capacity entails longer delivery times and straining production labor and machinery. Thus, **Node Utilization** in Listing 4.5 measures the extent to which a node employs its installed productive capacity after executing the demand fulfillment algorithm. **Average SCOR KPI** in Listing 4.6 is an example to calculate the average responsiveness of the SC nodes. This metric allows the estimation of the speed at which an SC provides products to the customer.

Listing 4.4: Order fulfillment metric.

```
PREFIX : <http://www.example.org/SENS#>.
SELECT  ?order (SUM(IF(REGEX(str(?x),"True"), 1, 0)) AS ?fulfill)
(SUM(IF(REGEX(str(?x),"False"), 1, 0)) AS ?notfulfill)
WHERE {
  ?order :isFulfilled  ?x.
```

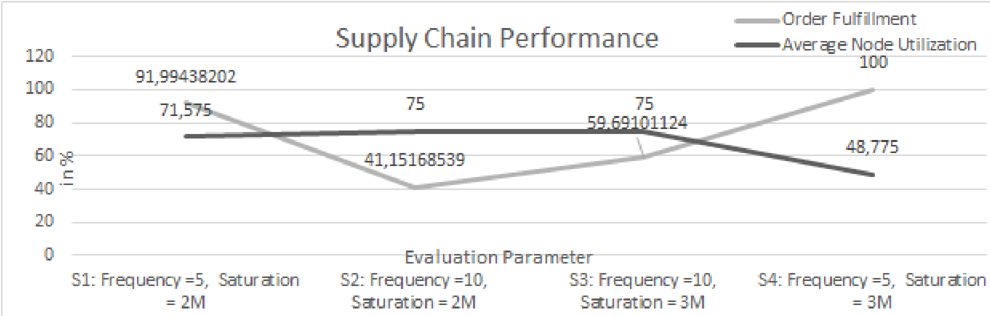


Figure 4.4: SENS knowledge-Graph performance evaluation with parameter variation: Customer_Demand_Frequency, Node_Capacity_Saturation .

```
}
```

Listing 4.5: Node utilization metric.

```
PREFIX : <http://www.example.org/SENS#>.
SELECT 100*?quant/?max
WHERE {
  ?sup :hasSaturation ?max.
  ?sup :hasCapacity ?cap.
  ?cap :hasQuantity ?quant.
  ?cap :hasTimeStamp 178.
}
```

Listing 4.6: Average SCOR KPI.

```
PREFIX : <http://www.example.org/SENS#>.
SELECT AVG(?res) AS ?Responsiveness
WHERE {
  ?supplier :hasResponsiveness ?res.
}
```

Parameter variation. We measure the performance of the SC under various experimental scenarios. We model this by changing the input parameters, we show the variation of behavior with different values for: Customer_Demand_Frequency, Node_Capacity_Saturation.

The graph in Figure 4.4 shows that the order fulfillment metric drops when the demand frequency doubles (on the x-axis S1-S2), which is a potential scenario during, e.g., the holidays season. Recovering with increasing saturation capacity can help the SC perform better, as we can see in the graph the surge in order fulfillment from S2 to S3 where Node_Capacity_Saturation increased from 2M to 3M. Moreover, we note that the node utilization is reduced when the Node_Capacity_Saturation increases. This result is logical as the nodes are not operating close to their production saturation. This is a required setup as it guarantees operational stability and constant delivery time. The average responsiveness is 85% and does not change with parameter variations.

Discussion of SENS

Including the SCOR model into SENS provides a standardized representation of SC processes and benchmarking KPI. Also, the E2E perspective brings an overall view of the SC partners and their relations and flow of supply and demand. Integrating these models using semantic artifacts facilitates the analysis of SC topology, interactions and operational behavior relying on standardized metrics.

We can extend SENS to optimize for additional node performance characteristics such as carbon footprint, service level, and price. This will enable extending the implemented supplier choice to include multi-factor-based decision making as explained by [157].

Also, we assume the nodes' characteristics to be constant throughout the simulation. As a result, the SENS parametrization is rigid to some extent, while real-life scenarios might impose some fuzziness. Thus, we propose to include a degradation function representing deterioration in behavior. For instance, the model should include delay functions for transit lead times or a variation of the SCOR KPIs in different operational conditions, e.g., to reduce responsiveness under high utilization.

In addition, we generate parameter values randomly or via user input. Alternatively, an interactive interface helps the user to tailor the values for each node individually and fine-tune the parameter space. Consequently, we can design restrictions on the parameters to reflect industry-specific characteristics. For instance, the semiconductors' production is layer after layer, thus the parallelization of the process to ramp-up production is not possible. Thus, Node_Capacity_Saturation parameter should remain constant.

SENS as a SC model is incorporated into the DR, a semantic model for semiconductor SCs.

4.2 Digital Reference: Vocabulary and Model for E2E Semiconductor Supply Chains

In the digitalization era, the semiconductor industry is at the heart of the digitalization efforts due to the role that semiconductor products play in electronics, IoT, and security systems. The semiconductor industry is complex with dispersed, globalized SC structures. The large range of customers with unstable demand for products and swiftly advancing technologies in a competitive market adds to the complexity in this domain. Moreover, the level of detail in the manufacturing process requires close supervision and monitoring. Thus, it is important to analyze the semiconductor industry from a holistic view while including the operational details.

Organizations rely on reference SC models or standard digital twins to observe, standardize, and benchmark their SCs. Existing E2E semiconductor SC models provide a description of the network topology. While manufacturing details are provided in operational models in an isolated manner as shown in the literature analysis subsection 2.2.6. Aligned with the four-layer division of the semiconductor SC by [87], we present the DR as a semantic model capturing E2E structures and flows with operational details of the semiconductor SC.

4.2.1 Digital Reference Description

The DR is an E2E representation of the semiconductor SCs that incorporates the structural description of the SC as well as the operational details and processes, e.g., semiconductor production, product lifecycle management and organization ontology [94]. The DR is an ontology written in OWL2, it contains annotations on OWL which detail the ontology modeler, the date of creation and any

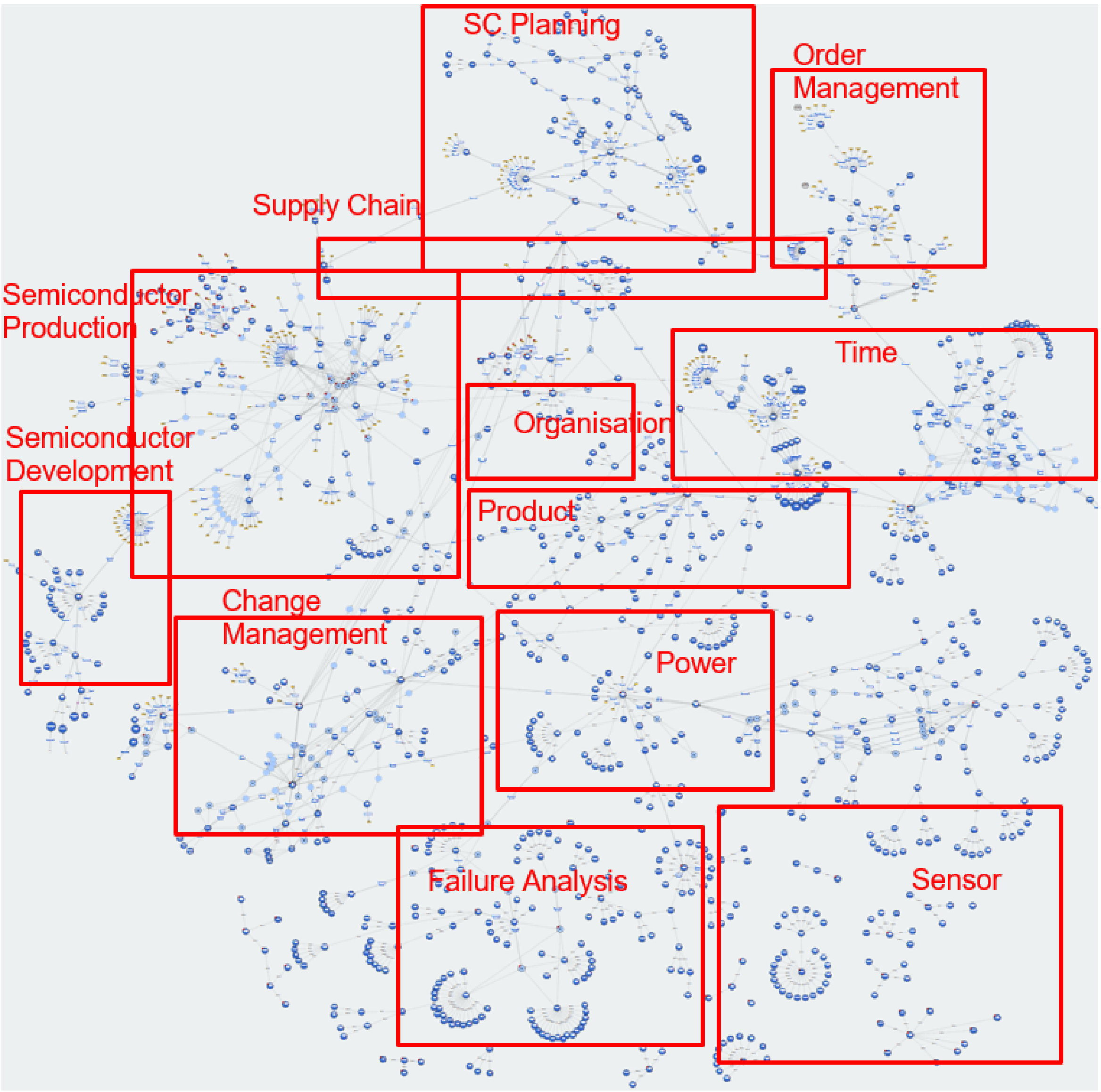


Figure 4.5: Digital Reference ontology model for semiconductor industry with incorporated subonotlologies such as semiconductors operations, order management and organization ontologies.

references to standard ontologies by the W3C [160], e.g., time and organization ontologies. As shown in Figure 4.5, the DR is organized in sub-ontologies that model different levels of granularity and details to include various aspects of the SC. For instance, the Supply Chain ontology represents a high level of detail about the structure and the flows in the SC. Yet, the Semiconductors Operations describe to details of the manufacturing process, locations, and tools. The DR is organized in sub-ontologies that model various SC concepts.

4.2.2 Digital Reference Subontologies

The following list provides a detailed description of the sub-ontologies in the DR:

- **Supply Chain Ontology:** We refer to SENS ontology in section 4.1 describing SCOR processes, E2E SCN network subsuming SC tiers, SC partners, i.e., nodes. We show the following exemplary triples: *Node manufactures Product* , *Node belongsToTier Tier*. The SCOR standard described by SENS includes the *Make* process detailing the production operations of semiconductors .

- **Semiconductors Operations:** This ontology defines the steps and entities involved in semiconductor production operations (of the *Make* SCOR process). It also describes facilities and manufacturing locations, e.g., *Wafer isManufacturedIn Fab Fab rdfs:subClassOf FrontEndFab*, *Fab hasLocation xsd:string*, *Wafer isDicedTo Chip*, *Chip isTestedIn Fab* and *Fab rdfs:subClassOf BackEndFab*

- **Failure Analysis:** This ontology describes the analysis process of failed products and clarifies the failure causes and mechanism, and provides feedback to the manufacturing and design process not only to prevent re-occurrence in the future but also to improve manufacturing and product quality. *Customer requests FailureAnalysis*, *Expert detects Failure*. The failure can be of different types, e.g., *Electrical rdfs:subClassOf Failure*. A failure is also localized *Failure hasLocation Fault Localization*. This ontology also describes the equipment needed to detect different types of failures.

- **SC Planning**: For the SCOR *Plan* process, this ontology defines different parameters, actions and entities involved in SC planning. In section 4.1, we detail the ontology for demand fulfillment, which consists of a major part of the SC planning to allocate capacities and production to fulfill customers' demand. Moreover, we introduce in section 6.2, an ontology describing the customer context information to allow a better understanding of the demand for production planning.

- **Order Management:** We describe the states and information used in order management process, part of the *Deliver* SCOR process. Customers communicate their orders with manufacturers via web portals or Electronic Data Interchange (EDI). SENS, in section 4.1, details the customer orders and the corresponding products and quantities. Moreover, in section 6.1, we detail the different lead and delivery times describing an order. Also, the order management ontology subsumes the SC responsible, e.g., Customer Logistics Management (CLM), for handling these orders, i.e., *CLM manages Order*.

- **Change Management:** This ontology defines the change to products processes within the business and manufacturing operations, i.e., *ChangeProject affects Product*.

- **Product:** This ontology describes the details of a semiconductor product. We distinguish two concepts. On the one hand, Production Product represents the end result of the manufacturing process and is identified by Production Nr. On the other hand, Sales Product represents a sellable product and is identified by Sales Nr. A thorough description of the products' details and identification is described in Master Data ontology section 5.1. Also, during the assembly process in the backend, products are packaged. The ontology in section 6.3 describes the packing information assigned to the product and the corresponding weight, height, and material.

- **Power:** Describes in detail the states and components that a chip needs to manage its power.
- **Sensor:** Defines the different actions, parameters, and states that a sensor can be or perform.
- **Semiconductors Development:** This ontology describes the different phases of a product lifecycle. The product is developed, i.e., designed, engineered, computed, virtually tested, simulated.
- **Organization**: We rely on standard ontologies defined by W3C Organization [160] to model the organization structure and decision-making chain in the SC .
- **Time:** We use W3C Time [161] to model temporal aspects needed for processes and sequences in production.

We rely on ontology merging tools such as Protégé [162] to bring together all the previously mentioned sub-ontologies in one ontology: the Digital Reference, publicly available on `https://w3id.org/ecsel-dr`. The scale of the DR continues to expand with the integration of further relevant ontologies, e.g., transit time ontology as described by [163].

4.2.3 Digital Reference Evaluation

We evaluate the DR in two methods. We check the correctness of the proposed ontology. Then, we address the aspects of the semantic model to cover the reference model requirements.

Digital Reference Correctness

We use ontology-based reasoning in HermiT to validate the structural and syntactic correctness of the DR. HermiT [164] [165] is the default reasoner in Protégé [166]. HermiT is an OWL 2 open source reasoner which provides the justifications for its inferences, as well as for the found inconsistencies.

Digital Reference as a Reference Model

[90] define the requirements for a reference model for the semiconductor SCs as shown in Table 2.3. In Table 4.3, we demonstrate the features of the DR that ensures it is a reference model. We can conclude that the DR is a reference model.

4.2.4 Digital Reference Application

The semiconductor industry is capital-intensive due to expensive equipment and the presence of rapid innovation cycles. As a result, companies in the semiconductor industry need to fiercely adapt their operations to such an evolving environment, and in turn, require their SCs to be highly resilient and agile. In order to overcome such challenges, simulation models are often used to analyze future scenarios, as well as to evaluate proposed changes or new concepts. However, for such a complicated domain, simulation models require high effort for modeling and computation. Reusability of models, using reference models or testbeds, potentially reduce modeling efforts. The DR, as a reference model,

No.	Requirement	Digital Reference
R1	Model a base system	Supply Chain ontology includes resources and corresponding features and behavior
R2	Include products	Product, Power, Sensor ontologies define the product and the detailed states and components
R3	Represent customers	Order Management ontology models how customers generate orders
R4	Demand information	Supply Chain ontology incorporates a demand fulfillment model and demand frequency parameter
R5	Planning	SC Planning and Semiconductors Operations ontologies detail the capacity allocation for production planning
R6	Information flow	Organization ontology determines the structure and entities for decision-making entities
R7	Control flow	Change Management describe planning and control instructions communicated to customers
R8	Understandable data structures	DR is written in OWL2 which specifies an XML serialization that models the structure of the ontology
R9	Performance measures	Supply Chain ontology incorporates SCOR KPIs

Table 4.3: Digital Reference satisfying all requirements for a semiconductors reference model.

provides a comprehensive representation of the domain with the required level of detail, which enables a smooth creation of simulation models.

4.3 Bridging Simulation And Ontology To Ease The Model Creation

Complex SCs, such as the semiconductors, require large modeling and computational burden for simulation. Semantic models provide a comprehensive representation of the domain that allows the modeler to better understand the domain thus, potentially reducing the time and effort to create the model. Despite the several similar approaches in the literature subsection 3.2.2, we note that the proposed work does not elaborate on the application of the ontology-based simulation in the industry but rather focuses on the theoretical approach. Thus, we present a concrete methodology aligned with existing work to enable the automation of the building process of a simulation model in the industrial context. We apply the approach in a use case from the semiconductors domain.

4.3.1 Methodology

We propose a methodology to use ontologies as a standard to develop and deploy simulation applications. Figure 4.6 shows the overall concept that includes simulation ontology, domain ontology and rule-based engine.

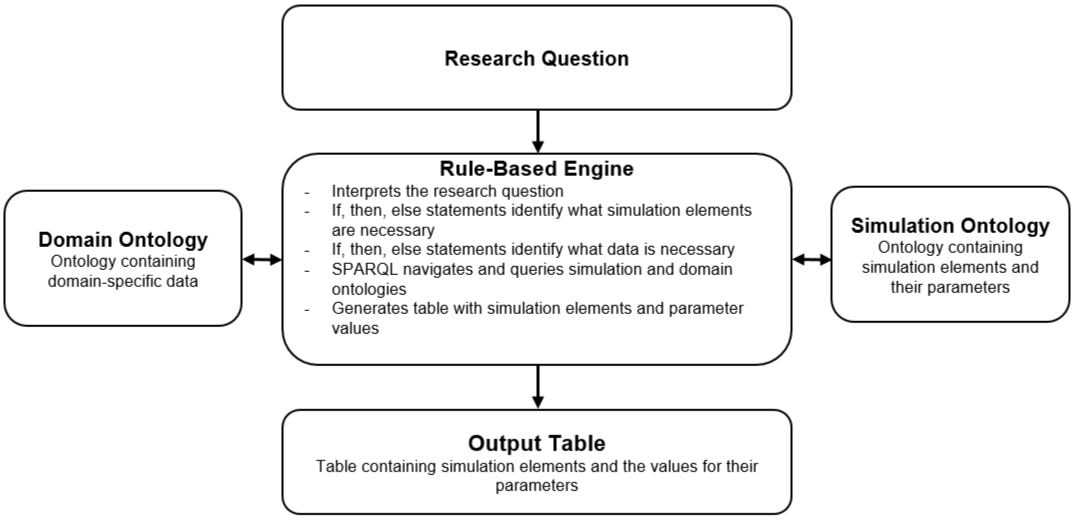


Figure 4.6: Overall concept of the ontology-based simulation methodology incorporating simulation and domain ontologies, and the rule-based engine. The input is the research question and the output is a table containing simulation elements.

Simulation Ontology

Ontologies provide explicit, uniform semantic descriptions of terms and concepts, which allows to correctly understand the flow logic and decision logic within the real-world processes being modeled [114]. Simulation ontology describes the simulation model constraints, which can be used to map specifications of real-world constraints that are found within the domain system descriptions. We rely on discrete event simulation as we can model with a high level of details and on a small level of granularity which enables the representation of SC operational problems as elaborated in Table 2.2 by [167] and [168]. We rely on DeMo simulation ontology, introduced by [169] and presented in subsection 3.2.2, as it models discrete event simulation artifacts. The models within this ontology capture discrete state changes via events. Through explicit descriptions of the concepts assumed in each of the DES world views, as well as the relationship between these concepts, DeMO enables sharing of these descriptions in an understandable language both by humans and machines.

Domain Ontology

According to [170], a domain ontology provides a semantic basis for requirements descriptions and achieves "lightweight semantic processing" in order to detect properties of requirements descriptions. [171] elaborate that in a domain ontology, the structure of a domain is described in terms of classes and properties. In fact, in the case of an ontology-based simulation, the domain ontology in question is an ontology describing entities, agents, data, inputs, outputs, and sub-processes involved in the processes to be simulated. Also, the domain ontology describes how these components relate to each other and interact within the domain.

In the scope of semiconductor SCs, the domain ontologies are divided in accordance with the four

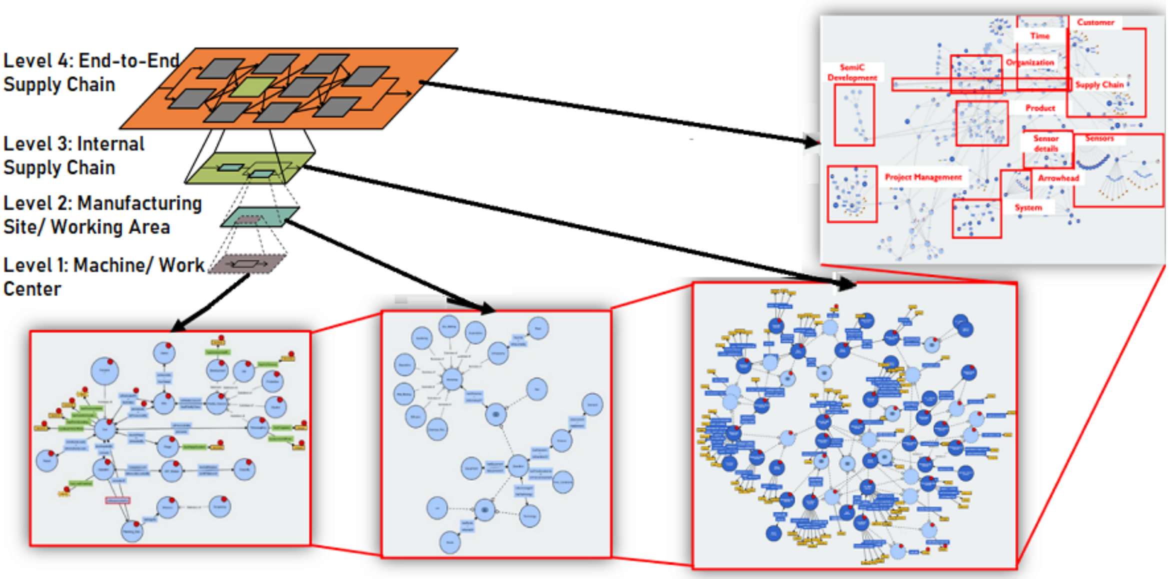


Figure 4.7: Simulation levels and corresponding ontologies where level 4 corresponds to the Digital Reference and level 1 depicts the lowest granularity in the semiconductors domain.

standard simulation levels [87]. Each level is initially represented by an ontology, merged into an abstract, higher-level ontology – the Digital Reference as shown in Figure 4.7. The Digital Reference facilitates the understanding of complex adaptable SCs and interactions between enterprises and eases the involvement of partner companies within the mutual vocabulary of terms used and functions depictions adopted. Additionally, the DR contains operational details to allow the creation of lower simulation levels.

Rule-Based Engine

The rule-based engine is the bridge between the domain and simulation ontologies and the simulation modeler. The purpose of the rule-based engine is to interpret a research question entered by the modeler, determine what simulation elements are required to investigate the question, and define what domain-specific data should be used in the simulation study. The rule-based engine prompts the user with a series of questions using the underlying if-then-else statement to narrow down the initial research question into a more precise one and identifies the information necessary to start building a simulation.

Once the research question has been deconstructed, the engine determines what needs to be simulated, e.g., tool, work center, factory, and SC. This is achieved by analyzing the objective, level of detail, and KPIs of the research question. Next, the engine determines what simulation elements, i.e., queue, delay, and split, are required to model the system via the simulation ontology. The simulation ontology also allows the engine to determine what parameters need to be defined using data from the domain ontology to build an accurate simulation. Through the interpretation of the research question and the information extracted from the simulation ontology, the engine builds a table showing what simulation elements are necessary and what parameters they will require for the simulation. Using

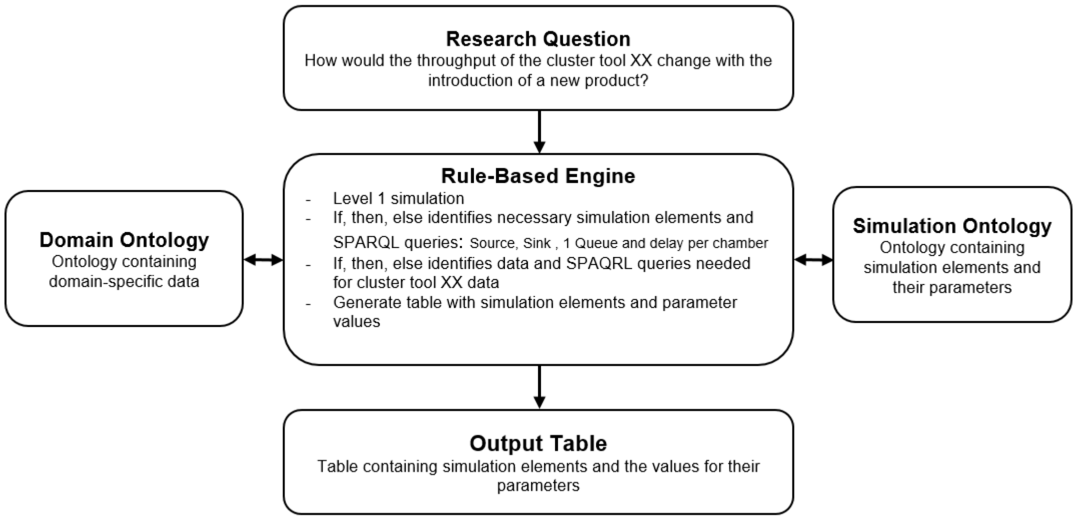


Figure 4.8: Overall concept of the use case. based on the research question, the rule-based engine identifies the necessary simulation elements from the simulation ontology and the values from the domain ontology to generate the output table.

SPARQL, the engine queries the simulation ontology to generate a list of parameters necessary for each of the elements.

The values of the parameters are determined through the domain ontology. Based on the details of the research question, the engine uses the domain ontology to identify values for the necessary parameters, i.e., capacities, throughputs, and maintenance schedules for each of the simulation elements. The engine leverages SPARQL queries with the domain ontology to determine the number of elements required and the proper values for their parameters. The values are then entered into the table of simulation elements and parameters for the user to build the simulation. After the engine finishes running, the user has a simple list of simulation elements and their parameters based on actual data to enable the quick and accurate assembly of a simulation model. The rule-based engine is a script written in python or java to execute the described tasks.

4.3.2 Use Case

We present an example of how the proposed approach applies to a use case for throughput analysis on a piece of semiconductor manufacturing equipment with the introduction of a new product. The specific piece of equipment being analyzed is known as a cluster tool that processes wafers. A cluster tool has several process bays that perform different steps in the manufacturing process and, depending on the specific product being produced, will have many repeated steps with different durations. The wafers arrive at the tool in lots, are loaded into the machine via a load port (which operates under vacuum pressure), and then the wafers proceed through the machine according to their recipe one by one. Once all wafers are processed, the lot is removed from the machine through an exit port.

Table 4.4: Simulation ontology describing the relations between main concepts such as the source, queue, delay, and sink.

Subject	Predicate	Object
Source	hasInterarrivalTime	InterarrivalTime
Source	hasAgentsPerArrival	AgentsPerArrival
Source	hasAgentType	Agent
Queue	hasAgentType	Agent
Queue	hasCapacity	Capacity
Delay	hasDelayTime	DelayTime
Delay	hasCapacity	Capacity
Delay	hasAgentType	Agent
Sink	hasAgentType	Agent

Rule-based Engine

The research question the user would pose to the engine would be formulated as follows: “How would the throughput of cluster tool XX change with the introduction of a new product?” By asking the user a series of questions with the underlying if-then-else-statements, the engine determines that a level 1 simulation is needed, the relevant KPI is throughput, and the data will need to be pulled for cluster tool XX. In this use case, we assume the rule-based engine is the system user supported by python snippets of code. The engine determines via SPARQL queries that a simulation would require a source, a sink, and that each chamber of the tool should be modeled as a queue followed by a delay.

Simulation Ontology

The simulation ontology would build off of DeMO. The subclass which would be of greatest interest for the example case would be the process-oriented model. The process activities of relevance for the example case would be the simulation elements source, queue, delay, and sink. Using SPARQL in Listing 4.7, the engine queries the simulation ontology to generate a list of parameters in Table 4.4 necessary for each of the elements. The data properties of the classes Source, Queue, Delay, and Sink are intended to address the parametrization of the model by defining the potential parameters or properties each building block might have.

Listing 4.7: SPARQL query to select building blocks, i.e., simulation elements of the model: source queue, delay, and sink details

```
SELECT * WHERE {
  ?subject ?predicate ?object.
FILTER (regex(str(?subject), 'Source') ||  regex(str(?subject), 'Queue')||
regex(str(?subject), 'Delay')||  regex(str(?subject), 'Sink'))
}
```

Domain Ontology

Using the domain ontology, the engine needs to determine the number of chambers cluster tool XX has, the current products being produced on it, the proportional loading of each product, and

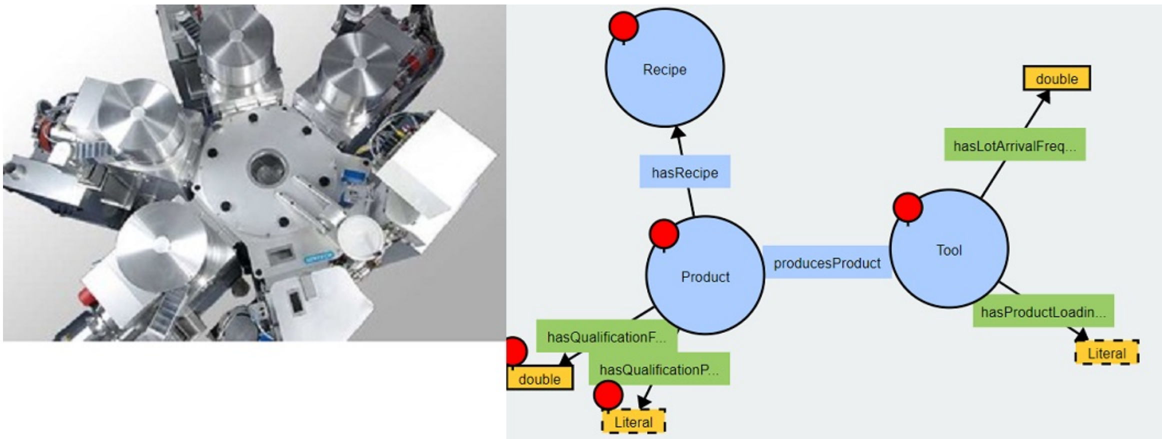


Figure 4.9: Left: cluster tool; Right: ontology representation of the tool.

their production recipes. The recipes determine which process bays are used and for how long. Consequently, the modeler relies on these values to determine the number of queues and delays representing chambers and the corresponding capacities. Additionally, the domain ontology provides information regarding the frequency of lot arrivals, lot size. These parameters and values define the source properties i.e., InterarrivalTime and AgentsPerArrival. The Figure 4.9 is an ontology depiction of the cluster tool by [172], it belongs to level 1 and is part of Digital Reference.

Listing 4.8: SPARQL query to select domain knowledge.

```
PREFIX : <http://www.example.org/DomainOntology#>.
SELECT * WHERE {
  ?source :produces ?agent.
  ?source :hasLotArrivalFrequency ?af;
          :hasProductLoadingProportion ?pp;
          :hasChamber ?x.
  ?agent  :hasQualificationFrequency ?qf;
          :hasQualificationProbability ?qp;
          :hasRecipe ?recipe.
FILTER (regex(str(?source), 'ToolXX') && regex(str(?agent), 'Product'))
}
```

The simulation modeler relies on the output of the SPARQL queries, i.e., the list of simulation elements and their parameters, to create the simulation model for the cluster tool.

4.3.3 Discussion

In the given example, the ontology can be expanded by merging ontologies representing other domains to add further machines and tools. Afterward, we can include details about a manufacturing site or a working area (level 2). Consequently, we may expand the scope to successively include further levels of simulation from manufacturing to SC operations.

4.4 Concluding Remarks

In this chapter, we presented semantic models to standardize and benchmark SCs. SENS is an integrated semantic SC model that enables standardization, behavioral analysis, and benchmarking. SENS leverages a well-defined ontology, SPARQL queries to include SCOR model artifacts, e.g., processes and performance indicators, as well as an end-to-end perspective to model SC partners and the flow of goods and materials. SENS-GEN is a highly configurable data generator that leverages the integrated SENS model to produce exemplary data based on input parameters and create a specific synthetic integrated instance of a SCN. Namely, SENS-GEN generates synthetic data to simulate SC behavior in controlled and designed scenarios. As a result, companies can rely on SENS and SENS-GEN to generate data for various simulated SCs. Additionally, we propose an extension to SENS, a reference semantic model for semiconductor SCs and SCs containing semiconductor: The Digital Reference. Semantic models are standardized representation of the domain, providing comprehensive details of the SCs. Modelers can rely on the semantic models to standardize the creation of simulation models to analyze SC behavior in a controlled environment and generate behavioral benchmarks. Ultimately, we deem that better modeling and simulation, enabled by ontologies and KG, will contribute to mastering more complex SC scenarios.

CHAPTER 5

Semantic Data Integration for Applied Supply Chain Management

Semantic data integration for SC applications enables integrated analysis capabilities, facilitates grasping, controlling and ultimately enhancing SC behavior. The content of this chapter is based on the publications [173], [174] to answer the following research question:

> RQ2: How can semantic data integration help make supply chains more efficient and resilient?

We introduce KnowGraph-MDM, a methodology for a knowledge-graph-based MDM, which relies on establishing a KG layer for effectively building a common understanding of the key business entities and semantic mappings from and to the original data sources. We also propose MARE a semantic disruption management and resilience evaluation framework to make SCs more resilient. MARE, aligned with existing DMP approaches, integrates heterogeneous data sources (e.g., production scheduling, order processing), covered by all DMP steps.

5.1 KnowGraph-MDM: A Methodology for Knowledge-Graph-based Master Data Management

MDM is an essential prerequisite for companies to make agile and correct decisions in their daily operations. MD reporting is about collecting and structuring MD information and translating it into a desired format or representation to assess ongoing business performance and accordingly make decisions. Understanding the schematic model of MD is important for reporting especially when accompanied by a mapping to the data layer. This mapping ensures efficiency in reporting as it prevents recreating unnecessary reports instead of re-using existing ones to retrieve desired MD. MD is a collaborative discipline that involves several SC stakeholders whose perspectives and inputs should be incorporated in the MD model.

As mentioned in section 2.1.2, traditional MDM approaches are limited in integrating enterprise information as well as meeting requirements, e.g., stakeholders' involvement for MD analysis and reporting. Research presented in subsection 3.2.4 shows that semantic-based MDM methodologies fulfill reporting requirements but are still rather limited in the inclusion of stakeholders. Therefore, we propose a step-by step approach establishing a KG layer for building a common understanding of

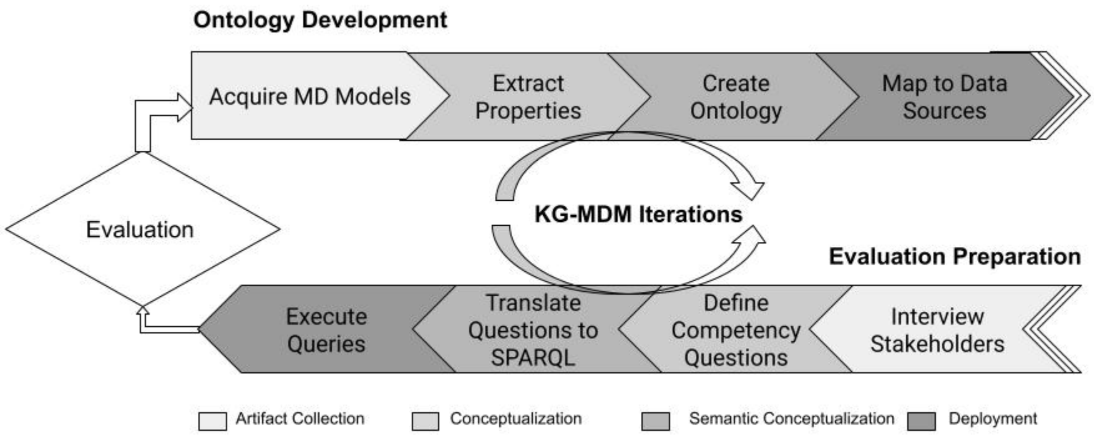


Figure 5.1: KnowGraph-MDM iterative methodology for knowledge-graph-based master data management with two streamlines to develop and evaluate the semantic model.

the key business entities and semantic mappings from and to the original data sources. KnowGraph-MDM (KG-MDM) relies on iterations to incorporate stakeholders' inputs allowing evolutionary development of the MD model. Thus, the ingestion and adoption of the new model increments among the stakeholders via the deployment in the organization. We apply the proposed approach in a use case from the semiconductors domain. The resulting KG depicts the core MD model that can be iteratively extended to incorporate different stakeholders' perspectives.

5.1.1 Methodology

The KnowGraph-MDM approach, outlined in Figure 5.1, defines a knowledge-graph-based MDM. KnowGraph-MDM creates a semantic MD model and obtains the consensus of different stakeholders. The approach consists of two streamlines, namely, ontology development and evaluation preparation. The first focuses on creating a KG representing the MD model, while the latter prepares an evaluation scheme for the model. The goal of the evaluation is to determine whether the model satisfies stakeholders' requirements or if further iterations are needed for extensions and modifications. An iteration consists of building or extending the KG as well as creating or modifying the evaluation scheme and the execution of the evaluation.

Ontology Development

In this streamline, we focus on exploiting various MD models to develop a Master Data KG (MDKG) that comprises the schema (ontology) and the data. The set of steps in this streamline are aligned with the methodology for ontology development proposed by [175]. First, we *Acquire MD Models*. This step is an artifact collection where we list existing MD models, whether they are conceptual, logical, or physical, e.g., databases, unstructured text, web pages, and entity-relationship diagrams. Once we identified the current models, we *Extract Base Properties*. This conceptualization phase is about choosing the focal points for the ontology model with concepts such as *Product* and *Customer*. The output of this step is the domain to be modeled in the schema or its extensions in later iterations. The next step is to *Create Ontology*, which is, in fact, a semantic conceptualization of the domain outlined in the previous step. We rely on the work of [176] and [177] to translate the extracted properties to a semantic model. Afterward, as the last step, the ontology concepts need to be *Mapped to Data Sources*. This makes the data semantically augmented and ready to be used for querying.

Evaluation Preparation

In this streamline, we prepare the evaluation for the MDKG created. The first step is to *Interview Stakeholders*. [178] proposes a methodology to identify domain experts and business stakeholders. We elicit the stakeholders' requirements and expectations from the MD model as part of the artifact collection. Define competency questions: The goal of the competency questions is to list the expectations of different stakeholders from the model, i.e., what concepts are covered and what data is retrieved. This stage is for the conceptualization of the model from the stakeholders' point of view. It is necessary to identify stakeholders' corresponding responsibilities, e.g., domain experts (manufacturing, engineering, logistics), MDM experts, and business stakeholders. Based on the interview outcomes, we *Define Competency Questions* as a conceptualization phase. This step is aligned with the work of [179] and [180] to employ competency questions and use them in evaluation of the ontology. After that, we *Translate Questions to SPARQL* , the semantic conceptualization allows answering competency questions on the MDKG via the *Execute Queries* step.

Evaluation and Iterations

The SPARQL queries reflecting the competency questions are communicated to the stakeholders. Their evaluation will guide the development process to either opt for more iterations or exit to the next phases of the model adoption. We conclude whether the MDKG is suitable or further modifications are required for either the KG, competency questions, or both. In fact, if the stakeholders judge the model to be incomplete, we reiterate the ontology development streamline. In that case, we do not need to recreate the competency questions, and we execute the SPARQL queries directly on the modified MDKG. However, suppose the questions are not enough to cover all stakeholders' definitions of MD entities of the domain in question. In that case, we redefine or extend the current competency questions and execute the queries on the current model. In some scenarios, the KG and the competency questions are not sufficient to satisfy all the stakeholders' requirements; therefore, we iterate on all the steps of both of the streamlines again. During the first iteration, all steps in the workflow have to be executed, whereas, in later steps, we can skip steps based on the output of the evaluation activities.

5.1.2 Use Case and Evaluation

The semiconductor industry is characterized by complex SC structures in a global network. Globally distributed manufacturing sites, processes, and information systems lead to scattered domain knowledge resulting in interactions between various agents and concepts. In this use case, we apply KnowGraph-MDM to create a unified view of scattered MD knowledge across 200 reports, including stakeholders' various inputs. We propose a set of technical details and tools while applying KnowGraph-MDM, but we reckon that there are other viable alternatives.

Ontology Development

Acquire MD Models. The primary goal of the first step is to choose data reports out of the large set of data reports that contain the essential key concepts for this prototype. This step is an artifact collection to list all existing MD models. In this use case, we assume that MD reports are representative of underlying dispersed MD sources. Thus, we rely on representative five Excel data reports acquired by two key figures, namely the *Number of Clicks* and the *Overlap of Data Fields*. The first determines the most frequently used data reports in 2019, 2020, and 2021. Reports are ranked by the number of clicks each year, and the corresponding overall priority of a report is calculated as the average value of the rank of all years. The higher the number of clicks within the three years, the higher the rank. To successfully identify the most relevant MD reports for this prototype, first, we choose fifteen MD reports with the most clicks. Subsequently, the second key figure determines the overlap of data fields. Each *data field* refers to an MD object. Therefore, an overlap of data fields reflects on its importance. As a result, the MD reports with the highest rank on the number of clicks and the most overlaps are the representative reports to be integrated into our model. Due to confidentiality reasons we refer to the reports as *Report A,B,C,D,E*.

Extract Properties. Next, in the step to *Extract Properties*, we conceptualize the acquired MD reports from the previous step. It can be assumed that the conceptual MD model within the semiconductor SCs is similar, i.e., not use case-specific. We identify two concepts that are primarily repeated in the reports, i.e., Production Product and Sales Product. On the one hand, Production Product represents the result of the manufacturing process and is identified by *Production NR*. On the other hand, Sales Product represents a sellable product and is identified by *Sales NR*. In fact, the relationship between *Sales NR* and *Production NR* is a 1:N-relationship, which means that one *Sales NR* is assigned to various *Production NRs*. The reason for this is, that from an MD-perspective, a *Production NR* has a higher granularity compared to *Sales NR*. This means, that *Production NR* contains more detailed data than *Sales NR*. For example, *Production NR* is adjusted even for minor changes in the manufacturing processes, whereas *Sales NR* changes only rarely and with major production changes.

Furthermore, *Basic Type*, *Package* and *DB BNR* are three important classes which are related to *Production NR*. The three classes represent the main stages of the semiconductor manufacturing process, namely the Front end, Die Bank, and Back end. The manufacturing process begins with the wafer fabrication and -probing in the Front-end. After successfully performing the first quality checks, the produced wafer is sent to the intermediate inventory storage Die Bank. Afterward, the chips are further processed in the Back-end. The wafer is cut, assembled, packed into different packages and tested in the Back-end part of the SC. Firstly, *Basic Type* is a product identifier that differentiates primarily technical and logistics aspects of the Front end- production stage. Secondly, *DB BNR* is a logistical identifier for chips lying in the Die-Bank facility of the semiconductor manufacturing process. Lastly, *Package* is the housing for a chip (or several chips) and provides electrical contacts and represents the Back end manufacturing processes. We show the extracted properties from *Reports A,B,C,D,E* in Table 5.1.

Create Ontology. An ontology refers to a semantic conceptualization based on the extracted properties. The extracted concepts, summarized in Table 5.1 are implemented in the ontology in triple format. We rely Protégé [181] as an ontology authoring tool. *Object Properties* connect classes

Properties for Sales NR		
Sales NR	assignedTo_ProductionNR	ProductionNr
Sales NR	assigned_SalesProductName	SPName
Sales NR	has_OrderablePartNr	OrderablePartNr
Sales NR	has_InternetSalesName	string
Sales NR	has_LastOrderDate	datetime
Properties for Production NR		
Production NR	has_Workroute	Workroute
Production NR	assigned_TechProduct	TechProduct
Production NR	contains	Package AND BasicType
Production NR	has_ProductStatus	string
Production NR	has_DeliveryTo_DistributionCenter	string
Production NR	has_ChipSequence	integer
Properties for Package		
Package	assignedTo_PackageAggregate	PackageAggregate
Package	has_TestLocation	TestLocation
Package	has_AssyLocation	AssyLocation
Package	packed_As	DieBank_BNR
Package	has_BackEndSegment	string
Properties for BasicType		
BasicType	pastOf_BasicTypeGroup	BasicTypeGroup
BasicType	hasSortLocation	Location
BasicType	has_FabLocation	FabLocation
BasicType	has_Thickness	integer
BasicType	has_ChipsPerWafer	integer
Properties for DieBank BNR		
DieBank BNR	contains_BasicType	BasicType
DieBank BNR	has_DispoPointDieBank	string

Table 5.1: Examples for extracted properties and classes from the master data reports to describe the Sales NR, Production NR, Package, Basic-Type, and DieBank.

to other classes, whereas *Datatype Properties* connect classes to data literals. For example, the Object Property *assignedTo_ProductionNR* connects the class *Sales NR* to the class *Production NR*. We implement a cardinality constraint that allows a specific minimum and a maximum number of values for that property. The object property between *Production NR* and *Sales NR* is a 1:N relation. Therefore the *min cardinality* is 0 and the *max cardinality* should be infinite. However, it is impossible to implement an unlimited number as max cardinality. Therefore, we used a large number. The Datatype Property *assigned_SPName* connects *Sales NR* to the data literal string. In Figure 5.2, we show an extract of the resulting MD ontology.

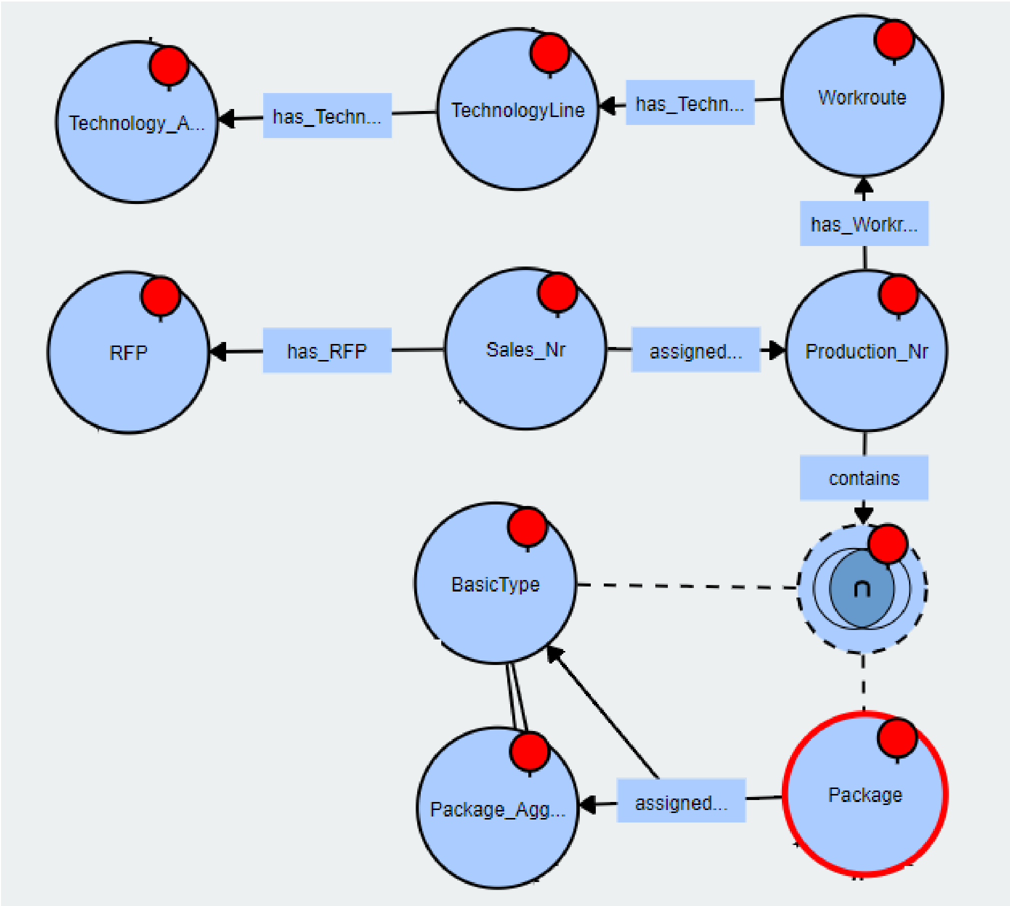


Figure 5.2: Extract of the visualized master data ontology including Sales Product and Production Product.

Map to Data Sources. Connecting the data and the ontology is part of the step *Map to Data Sources*. Based on its availability and technical proficiency, we create instances from the data sources in triple format by using the Corporate Memory Tool (CMEM) developed by Eccenca, a German provider for Enterprise KG solutions [182]. However, other tools exist, such as the Protégé built-in plugin Cellfie [183]. Finally, created instances from the data sources are defined in triple format and deployed on the *KG*.

Listing 5.1: Triples example

```
PREFIX : <http://exampleURI/KG_MDM#>
:SP001652718 :assignedTo_ProductionNR :PP12804.
```

The triple above shows an example from the output KG containing triples from the MD data sources in this use case. The subject, an instance of the *Sales NR*, SP001652718 is assigned to the object, an instance of *Production NR*, PP12804 by the predicate *assignedTo_ProductionNR*. The output of the first streamline is the MDKG, used as input for the second streamline.

Conceptual Model	**1:** Which Data Fields are contained in a specific Data Report? **2:** In which data report is a specific set of Data Fields contained? **3:** What is the relationship between Data Fields? **4:** What is the cardinality restriction describing the relationship between different Data Fields?
Data Model	**5:** Which are the Business Segments and Product Groups of a specific Product? **6:** What is the reached Milestone and Product Status of a specific Product?

Table 5.2: Output of stakeholders' interview as list of competency questions covered by the ontology model.

Evaluation Preparation

After creating the MDKG, we move on to the second streamline to evaluate the semantic implementation. Based on SC stakeholders' input, we expand and modify the model.

Interview Stakeholders. In this step, we collect the stakeholders' requirements and expectations of the prototype as artifacts. After discussion with several MD experts, we distinguish the requirements of the conceptual and data models. Firstly, the requirements of the conceptual model cover the description of the company's key business objects and the respective relationships on a schematic level. Secondly, the requirements of the data model cover the physical data, which is integrated into the MDKG. We summarize the output of the stakeholders' interview in table 5.2.

Define Competency Questions. We translate the stakeholders' requirements and expectations of the model into competency questions in the next step. Competency questions are defined in natural language to determine whether the stakeholders' expectations and requirements can be met by using the model.

Translate Questions to SPARQL and Execute Queries. The next step is to translate the defined competency questions into SPARQL as semantic conceptualization and execute the queries as deployment. We translate the competency questions manually to SPARQL. Alternatively, we can use frameworks, such as AutoSPARQL [184], PAROT [185], to automatically translate from natural language to SPARQL. The SPARQL queries are executed in the Java-based Integrated Development Environment Eclipse with the Apache Jena Fuseki Package. For conciseness, we show some of the SPARQL queries and sample results to show the deployment of MDKG. The rest of the queries are Appendix C.

CQ 1: Which Data Fields are contained in the data report A?

Data Report A is about product details, so this query returns all data fields contained about products, e.g., technology group, product line.

CQ 2: In which data report are the Data Fields [SP_Nr, PR_NR] contained? – AND Relationship

Competency Questions 2.1 and 2.2 both contain the same general question. However, the difference is the query condition. In CQ 2.1, an *AND* relation is queried, which means that a data report should

Competency Question

```
PREFIX smi: <http://exampleURI/KG_MDM#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?value ?object
WHERE {
{
  ?value rdfs:domain smi:ReportA.
}
UNION {
  ?subject rdfs:range smi:ReportA.
  ?subject rdfs:domain ?object.
}
}
```

SPARQL Result	value	object
	has_Technology	TechnologyAggregate
	belongsToPoductLine	ProductLine

Table 5.3: SPARQL query and results of competency question 1: "Which Data Fields are contained in a specific Data Report?".

Competency Question

```
PREFIX smi: <http://http://exampleURI/KG_MDM#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT  ?var1 ?var2 ?report
WHERE {
  ?relation rdfs:domain ?var1.
  ?relation rdfs:range ?report.
  ?relation rdfs:domain ?var2.
  ?relation rdfs:range ?report
FILTER(regex(str(?var1),'Sales_Nr') &&
regex(str(?var2),'PR_Nr') ).
}
```

SPARQL Result	Var1	Var2	Report
	Sales_Nr	PR_Nr	ReportC

Table 5.4: SPARQL query and results of competency question 2.1: "In which data report is a specific set of Data Fields contained? – AND Relationship".

be found that contains the data fields Sales NR *AND* PR Nr. The following result shows that both data fields are contained in Report C. In CQ 2.2. an *OR* relation is queried. In this case, a data report should contain Sales NR *OR* PR Nr.

CQ 3: What is the relationship between Data Fields SP_Nr and PL? The result of the third competency question in Table 5.5 shows the classes and relations which are in between the data fields SP_Nr and PL.

CQ 6: What is the reached Milestone and Product Status of Sales Product Name 10804DA?

Competency Question

```
PREFIX smi: <http://http://exampleURI/KG_MDM#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT  *
WHERE {
  ?relation1 rdfs:domain smi:SP_Nr.
  ?relation1 rdfs:range ?class1.

  ?relation2 rdfs:domain ?class1.
  ?relation2 rdfs:range ?class2.

  ?relation3 rdfs:domain ?class2.
  ?relation3 rdfs:range ?class3.

  ?relation4 rdfs:domain ?class3.
  ?relation4 rdfs:range smi:PL.
 }
```

SPARQL Result	**Relation1**	**Class1**	**Relation2**	**Class2**
	has_RFP	RFP	has_FGR	FGR
	Relation3	**Class3**	**Relation4**	
	has_HFG	HFG	has_PL	

Table 5.5: SPARQL query and results of competency question 3: "What is the relationship between Data Fields?".

The results of CQ 6 in Table 5.6 demonstrate, that Sales Product Name 10804DA has the *Product Status* active and the *Milestone* M9. The Product Status M9 implies that the product is delivered to the customer in an unrestricted manner. With this question, we emphasize the importance of mapping the conceptual and data models of MD. The understanding of the domain incorporated in the conceptual model implies the product life cycle concepts. The mapping to the data model allows to retrieve the data that maps to this concept.

Iterations

The streamlines *Ontology Development* and *Evaluation Preparation* are iterative. In total, three iterations were run to end up with comprehensive ontology and competency questions that cover the defined model requirements of the stakeholders. Within the first run, all steps are performed. Subsequently, further iterations revisit specific steps.
In the *first iteration*, we revolve and refine the first draft of the ontology. Ontologies must reflect the domain from an MD stakeholder perspective and accurately enable the data mapping. By defining basic competency questions, translating them to SPARQL, and executing the queries, we obtain an ontology that is a reasonable basis for further iterations. However, after the evaluation, stakeholders recognize that the cardinality restrictions are required.
In the *second iteration*, we first revisit the step to create ontology and extend the current ontology to the defined stakeholders' model requirements, i.e., data-field-containing data reports and cardinality

Competency Question

```
PREFIX smi: <http://http://exampleURI/ KG_MDM #>
SELECT  ?sp_name ?p_status ?milestone
WHERE {
  ?sales_Nr smi:assigned_SPName ?sp_name.
  ?sales_Nr smi:assignedTo_FP ?fp_Nr.
  ?production_Nr smi:has_ProductStatus ?p_status.
  ?production_Nr smi:reached_QStatus ?milestone.
FILTER(regex(str(?sp_name),'10804DA')).
}
```

SPARQL Result	SP Name	P Status	Milestone
	10804DA	active	M9

Table 5.6: SPARQL query and results of competency question 6: "What is the reached Milestone and Product Status of a specific Product?".

restrictions. Subsequently, we translate stakeholder input into competency questions and SPARQL queries and execute the query on the redefined ontology. We include domain experts' knowledge by adding specific requirements, such as details about the engineering process of products into the model. Thus, after the second iteration, we show that model requirements can be fulfilled by executing the SPARQL queries on the current MDKG.
Lastly, in the *third iteration* we make minor changes in the model and readdress the steps to create ontology, translate questions into SPARQL and execute queries accordingly. Finally, the results fit the model requirements of the stakeholders. The current state of the model enables the sales and marketing stakeholders to link products with applications and systems. Consequently, semiconductors sales products are more reachable and accessible.

5.1.3 Discussion

Knowgraph-MDM ensures, through the link between physical and conceptual models, the interoperability between business and IT, which is an important factor in establishing a successful MDM as defined by requirement **R1** described in subsection 3.2.4. By integrating conceptual and data MD levels, KnowGraph-MDM supports integrated analysis and reporting of different MD aspects. KnowGraph-MDM relies on iterations to incorporate stakeholders' inputs allowing evolutionary development of the MD model **R2**. Moreover, KnowGraph-MDM is aligned with existing ontology modeling approaches and incorporates modeling and evaluation steps in one compact methodology. This combined step-by-step approach eases the deployment of KnowGraph-MDM in SC systems and makes semantic models reachable by users who are not familiar with SW artifacts. However, KnowGraph-MDM is limited as it does not specify clearly an approach for stakeholder choice neither for the evaluation preparation phase nor for the evaluation itself.We propose that for each iteration, the choice of stakeholders is made based on specified requirements. Thus, we can facilitate collaborative ontology development by various stakeholders and provide continuous feedback loops that systematically reflect and change the model.

5.2 MARE: Semantic Supply Chain Disruption Management and Resilience Evaluation Framework

A vast share of enterprises rely on a Disruption Management Process (DMP) to monitor, model, assess, and recover from disruptions. The implementation of recovery strategies restores the SC to its pre-disruption state. Namely, the ability to both resist disruptions and recover the operational capability after disruptions occur is defined as SC Resilience [186].

Existing approaches address core DMP aspects but still in an isolated form, hence, limiting integrated SC behavioral analysis. Compared to previous work, our main contribution in this paper is MARE. MARE is a semantic disruption management and resilience evaluation framework, to integrate data covered by all DMP steps Monitor/Model, Assess, Recover and Evaluate.

MARE leverages a disruption ontology to model disruptive events and a KG to represent specific disaster scenarios and the entailed effect on the SC. MARE includes production scheduling data and disruption KGs to detect the implication of the disruption on the SC operations during the assessment phase. Thus, MARE implements SPARQL-based recovery strategies to resolve the impairment caused by the disruption. Moreover, MARE incorporates a semantic evaluation framework to quantify the effect of recovery in terms of cost and delay on the SC. Based on the evaluation results and the recovery behavior analysis, SC stakeholders potentially make decisions to redesign the SC or establish new operational strategies ensuring a more resilient SC.

5.2.1 Methodology

In this section, we describe our semantic disruption management and resilience evaluation framework, MARE. Moreover, we elaborate on MARE's semantic artifacts, i.e., ontologies, KGs and SPARQL to implement the DMP. As shown in Figure 5.3, the DMP starts with **Monitoring** and **Modeling** SC disruptions. This phase is to discover the event that disrupts the SC and to create a semantic model incorporating the disruption's attributes, e.g., severity, cause, and duration. We rely on the *Disruption Ontology* model, where the information is represented in the form of RDF triples, to establish a common understanding of a disruption event. Therefore, we create a specific instance of a disruption event, i.e., *Disruption KG*. The output of the *Monitor/Model* process step, the *Disruption KG*, is used in the following step to assess the effect of the disruption on the SC.

The target of an SC is to fulfill end-customers' demand. Namely, SC planning defines a scheduled capacity allocation for products among production facilities as well as the needed parts among suppliers, i.e., *Supply Plan*. In previous work section 4.1, we devised a semantic model for demand, production scheduling data and corresponding supply plan as follows:

- **Demand**: SC demand is represented by the triples of the following form *Customer makes Order*. An order includes details about the product, delivery time and quantity: *Order hasProduct Product*, *Order hasDeliveryTime xsd:dateTime* and *Order hasQuantity xsd:integer*. Based on the customer segmentation paradigm, customers are given a priority, entailing a certain sequence in demand fulfillment, i.e., *Customer hasPriority xsd:integer*.

- **Supply Plan**: A supply plan is defined as the allocation of demand for parts among suppliers or the allocation of demand for products among production facilities [187]. *Order hasSupplyPlan Plan* and *Plan needsPartner Partner* describe the needed SC partners to fulfill this order. Each partner is responsible for providing a product, i.e., << *Plan needsPartner Partner* >>

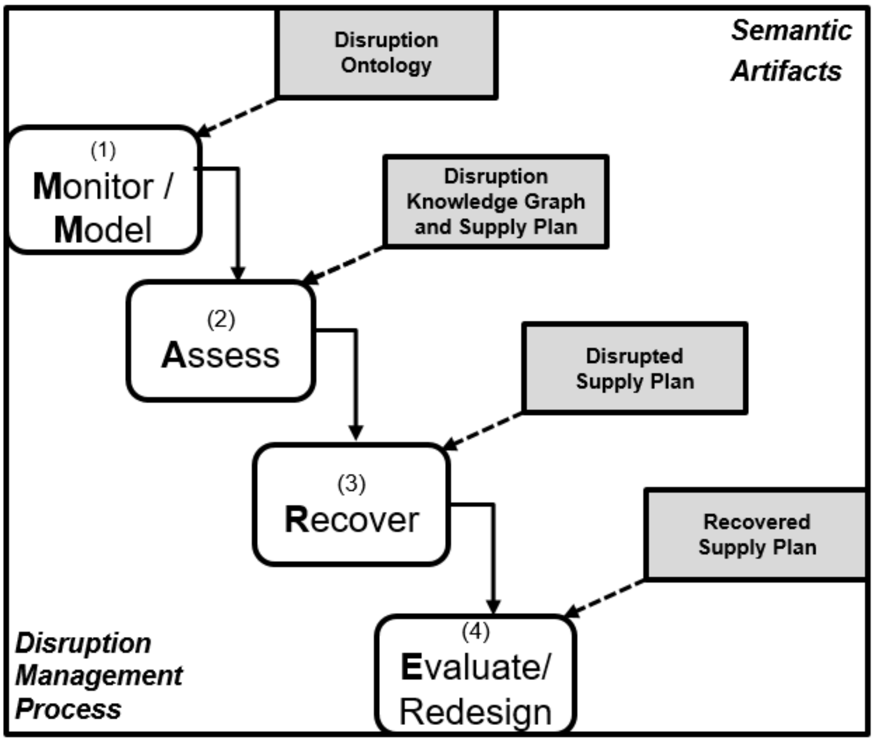


Figure 5.3: Overview of MARE semantic disruption management and resilience evaluation framework with semantic artifacts to Model, Assess, Recover and Evaluate disruptions.

getsProduct Product at a certain time *hasTimeStamp xsd:date*. The mentioned product can either be the final product or intermediary parts used to manufacture the final product. The quantity and the price are modeled using *hasQuantity xsd:double* and *hasUnitPrice xsd:double*

Disrupted SC partners potentially cannot fulfill their role in the plan, which affects the whole SC performance. Therefore, during the disruption **Assessment** phase, we leverage queries adhering to the W3C SPARQL standard to identify affected SC partners that are located in the same regions as the disruptions and who participate in the supply plan at the same time of the disruption. In this process step, we integrate data sources from production scheduling (*Supply Plan*) and disruption models (*Disruption KG*) to output the *Disrupted Supply Plan*.

The following step in the DMP is applying **Recovery** strategies to attempt a return to the pre-disruption performance of the SC. In this phase, we rely on SPARQL endpoints to integrate data from production scheduling, order processing, inventory management, and suppliers assignment in order to find alternative allocations for the disrupted plans.The output of this step is one or more proposed *Recovered Supply Plans* that include the updated scheduled allocations.

The last step of the DMP is to *Evaluate* the SC recovery performance. We propose a resilience **Evaluation** framework based on SPARQL queries to examine the time and the cost entailed by the *Recovered Supply Plan* and required for the SC to return to the pre-disruption state. In fact, SC stakeholders rely on this evaluation to potentially identify needs to redesign SC or apply new operational strategies, e.g., supplier diversification.

5.2.2 Supply Chain Disruption Modeling and Assessment

In this section, we present the first two steps of MARE to model and assess the effect of monitored disruptions on the SC.

Modeling Disruption

SC disruptions are modeled for a better understanding of the unexpected events, their cause, and effects. Disruption models help quantify and assess disruptions and study interdependencies between them.

Disruption Ontology. We propose the *Disruption Ontology* shown in Figure 5.4 to establish a model for disruptive events. The ontology is based on RDF, where the information is represented in triples. First, a triple of the following form *Disruption hasCause Cause* describes the main cause that led to the disruption. In fact, [188] classifies disruption causes as internal and external. The first is caused by events happening within internal boundaries and the business control of the organizations, e.g., malfunctioning of a machine or inventory corruption. The latter is driven by events either upstream or downstream in the SC, e.g., insufficient supplier capacity, interruptions to the flow of product, or significant increase/decrease in demand.

Moreover, disruptions impact various SC scopes, e.g., production, logistics, inventory [47]. This, is reflected by triples of the form: *Cause hasScope xsd:string*. Additionally, the structure *Disruption hasSeverity xsd:string* incorporates financial losses caused by the disruption and their effect on the reduction or elimination of the production quantities. Further, disruption events can be of short or long duration. We use the following triple representation to model the disruption beginning and end *Disruption hasBeginDate xsd:date* and *Disruption hasEndDate xsd:date*. Also, we use *Disruption hasLocation xsd:string* to represent the geographical location where the disruption occurs. We rely on geo-coordinates system to resolve locations using the properties *hasLongitude, hasLatitude*.

In fact, classifying the modeled characteristics of the disruption enables SC stakeholders to determine suitable recovery strategies for this event. For example, in case of an external disruption due to the lack of a supplier's capacity, the recovery means can be to find an alternative supplier. Whereas, to recover from an internal malfunctioning machinery within an own facility, one needs to fix it by retrieving spare parts from a machine of the same brand.

Instantiated Examples. The proposed disruption ontology incorporates disruption attributes to create a specific instantiation of a disruption event, represented by the *Disruption KG*. We present in Table 5.7 various examples from past events to highlight possible variations in disruptions in terms of cause, scope, location, duration, and severity.

:Disr1 is an example of capacity scarcity caused by labor shortage after a COVID-19 outbreak that led to a complete shutdown of production lasting four days. *:Disr2* shows a very short disruption, as the fire lasted for 10 minutes and the physical damages were minimal, i.e., the severity is low. Further, the medium contamination described by *:Disr3* affected not only the production plant but also the stockpile inventory.

Moreover, due to a halt in maritime transportation mode caused by a blockage in the Suez Canal, Sony sales dropped from 70,000 a week to around 6,000, i.e., *:Disr4*. In fact, supply shortage includes scarcity in raw materials or any event (bankruptcy, over-demand) that leads to a reduction

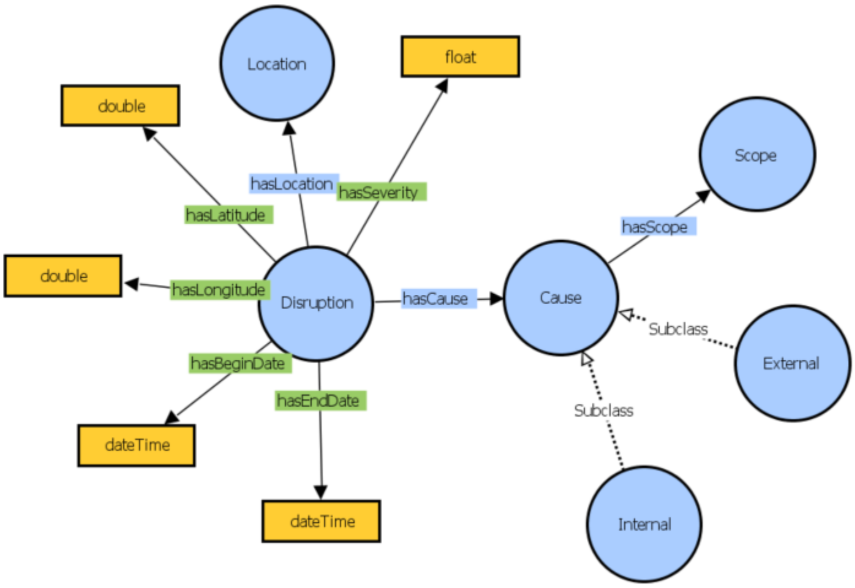


Figure 5.4: Overview on the core concepts of the disruption Ontology for modeling disruptive event characteristics such as cause, severity and location.

or discontinuation in supply. In 2020, due to the COVID-19 pandemic, the automotive industry suffered from a substantial drop in demand that led to slowing their semiconductor orders. Meanwhile, the semiconductor manufacturers faced a significant increase in demand due to the rising need for personal computers, servers, and equipment while their own facilities were shutting down because of COVID-19 outbreaks [189]. For instance, *:Disr5* representing over-demand, halted production and unstable orders led BMW to recognize a loss of 30,000 units in production so far in 2021. This disruption has an undefined end date. Similarly, *:Disr6* models the missing color pigments produced by factories in Japan affected by the Tsunami in 2011. *:Disr6* has medium severity since car manufacturers limit ordering vehicles only in specific shades.

Disruption Assessment and Effect

After identifying and modeling the disruption, the following step is to assess the impact. SC disruptions potentially hurdle SC entities from achieving operational goals, i.e., fulfilling end customers orders. We leverage data from production scheduling and order processing, i.e., *Supply Plan* along with the modeled disruption from the previous step, i.e., the *Disruption Knowledge-Graph*.

The first step while assessing the disruption effect is to identify the SC partners that are part of a supply plan yet fall within the disruption location and time frame. Listing 5.2 [1] retrieves and labels SC partners and corresponding *Disrupted Supply Plan*. Also, the effect of the disruption is defined by how many supply plans are affected. We insert *Disruption affectsPlan xsd:integer*, i.e., the count of disrupted plans identified in Listing 5.2.

[1] For simplicity, the query is just using a standard longitude/latitude matching, but in our implementation, we actually implemented a geospatial rectangular containment matching between supplier and disruption locations.

Examples / Triples	:Disruption	:hasCause	:hasScope	:hasLocation	:hasBeginDate/ :hasEndDate	:hasSeverity
Closing of Amazon warehouse due to the COVID-19 outbreak (Srinivas and Marathe, 2021)	:Disr1	Capacity Shortage	Production	Kentucky, USA	26.03.2020/ 01.04.2020	Medium
Fire in the Philips semiconductor plant (Sheffi and Rice Jr, 2005)	:Disr2	Fire	Production	Albuquerque, USA	17.03.2001/ 17.03.2001	Low
Discovery of Vesivirus2117 in Genzyme in a bioreactors in plant and its stock inventory (Tomlin and Wang, 2011)	:Disr3	Contamination issues	Inventory	Massachusetts, USA	16.06.2009/ 19.06.2009	Medium
Halt on the Suez Canal Sony gaming lead to supply shortage (Romeike and Hager, 2020)	:Disr4	Block in transportation mode	Logistics	Suez Canal, Egypt	15.10.2004/ 01.12.2004	High
Production halt at BMW in some plants due to semiconductor chip shortage (Reuters, 2021)	:Disr5	Supply Shortage	Production	Germany	01.01.2021/ Unknown	High
Pigment shortage hits Auto Makers after Tsunami (Canis, 2011)	:Disr6	Supply Shortage	Production	Japan	01.01.2011/ 28.02.2011	Medium

Table 5.7: Disruption examples and corresponding triple representation with varying cause, scope, location, begin date, end date, and severity; ':' is the ontology prefix.

Listing 5.2: Identify disrupted partners.

```
PREFIX : <http://www.example.org/MARE#>.
INSERT {
  ?plan :isDisrupted 'True'.
  ?disruption :affectsPartner ?partner.
  <<?plan :needsPartner ?partner>> :isDisrupted 'True'.
 }
WHERE {
  <<?plan  :needsPartner ?partner>> :hasTimeStamp ?t.
  ?partner :hasLongitude ?long.
  ?partner :hasLatitude ?lat.
  ?disruption :hasLatitude ?latitude.
  ?disruption :hasLongitude ?longitude.
  ?disruption :hasStartTime ?start.
  ?disruption :hasEndTime ?end.
FILTER (?t>=?start && ?t<?end && ?longit=?long && ?lat=?latitude)
 }
```

The second step is to size the effect of the disruption on the disrupted SC partners. The severity of the disruption determines the impact of the event on the partner's capacity to fulfill the supply plan. For simplicity, we model the severity as a numerical factor that shows the reduction in production capacities caused by the disruption. As shown in Listing 5.3, the pre-disruption allocated quantity is reduced by the severity factor. The difference between the original and the reduced quantities represents the quantity to be supplied or produced by alternative partners and means.

Listing 5.3: Determine disruption impact.

```
PREFIX : <http://www.example.org/MARE#>.
SELECT * WHERE {
  <<?plan :needsPartner ?partner>> :isDisrupted 'True';
  :getsProduct ?product;
  :hasTimeStamp ?t;
  :hasQuantity ?q.
  ?disruption :affectsPartner ?partner.
  ?disruption :hasSeverity ?factor.
BIND (?q*?factor AS ?reduced). BIND (?q-?reduced AS ?toRecover)
}
```

5.2.3 Recovery and Resilience Evaluation

After modeling and assessing the disruption effect on the supply plans, the next steps in the DMP are to implement recovery strategies and evaluate the SC resilience and recovery performance.

Supply Chain Recovery

In this section, we describe the implementation of the third step of MARE, i.e., Recovery. Recovery strategies are actions applied to regain the pre-disruption state of the SC, capable of delivering products to customers on time while minimizing the cost. By integrating data sources about inventory management, resources procurement, supply management, and logistics, we aim to recover disrupted supply plans. We present recovery strategies that rely only on the change in the SC planning and do not require any physical modification in the industrial process as the latter is highly dependent on the industry. For instance, increasing production capacity or allowing faster production are not realistic in capital intense or complex industries like semiconductor production. We propose the following SPARQL-based recovery strategies capable of adapting the supply plan depending on the disruption cause and scope. For all the following queries, we assume the recovery is for Product P, at time T in quantity Q.

S1: Strategic Stock. is defined as a stockpile of inventory that can be used to fulfill demand during a disruption [190]. Listing 5.4 verifies if the partner has strategic stock and returns the required price. We use inventory management data sources to implement this strategy. In fact, storing the strategic stock entails costs for warehousing, labor, and insurance.

Listing 5.4: Strategic stock strategy.

```
PREFIX : <http://www.example.org/MARE#>.
SELECT * WHERE {
  :Partner :hasStartegicStock ?stock
  ?stock :hasTimeStamp :T.
  ?stock :hasQuantity ?q.
  ?stock :hasPrice ?price.
  ?stock :hasProduct :P.
FILTER (?q>= Q)
}
```

S2: Alternative Shipment. in case of a disruption affecting the transport mode, e.g., flights, trains, a company can switch to another shipment mode to deliver products. The query in Listing 5.5 retrieves the shipment modes employed by a partner and the entailed costs caused by the change of transportation modes, usually incorporated in logistics data sources [188].

Listing 5.5: Alternative shipment recovery strategy.

```
PREFIX : <http://www.example.org/MARE#>.
 SELECT * WHERE {
  :Partner :hasTransportMode ?mode.
  ?mode :hasCost ?cost.
 }
```

S3: Delayed Recovery. this recovery strategy consists of verifying the status of the disrupted partner if it can deliver slightly later than planned. Listing 5.6 checks for five days after the planned delivery time if an SC partner has enough capacity, lower than saturation, to fulfill the plan. In fact, small delays in deliveries can mitigate financial losses due to disruption [191]. Delays greater than five days (a week) potentially lead to fines of great amounts. Production management and scheduling data sources incorporate data about the continuous state of capacity production.

Listing 5.6: Delayed recovery strategy.

```
 PREFIX : <http://www.example.org/MARE#>.
 SELECT * WHERE {
  :Partner :hasCapacity ?cap.
  :Partner :hasCapacitySaturation ?sat.
  ?cap :hasProduct :P.
  ?cap :hasPrice ?price.
  ?cap :hasTimeStamp t_future.
  ?cap :hasQuantity ?q.
FILTER (?sat >= ?q + Q && t_future< T+5)
}
```

S4: Alternative Supplier. this strategy applies in case of an external disruption that hinders the supplier from providing the required products at the time included in the supply plan. In fact, [192] elaborates that suppliers have production flexibility that allows them to deliver a contingency quantity in case other suppliers fail. However, the alternate source of supply can be more expensive than the firms' primary suppliers, but it is deemed necessary in order to recover the disrupted supply plan [193]. To reduce purchasing prices and benefit from the high performance, suppliers that are capable of supplying the same products are exchangeable [153]. We model this via the property *hasGroup*. Listing 5.7 shows the query to find alternative, exchangeable suppliers that have the capacity (lower than saturation) to provide the same intermediate products or materials at the same time as the disrupted supplier. We rely on data from supply management and resources procurement to make decisions about suppliers belonging to the same group and their capacities.

Listing 5.7: Alternative supplier recovery strategy.

```
PREFIX : <http://www.example.org/MARE#>.
SELECT * WHERE{
  :Partner :hasGroup ?group.
  ?supplier :hasGroup ?group.
  ?supplier :hasCapacity ?cap.
  ?cap :hasProduct ?p.
  ?cap :hasQuantity ?q.
  ?cap :hasPrice ?price.
  ?cap :hasTimeStamp :T.
  ?supplier :hasCapacitySaturation ?sat.
FILTER ( ?sat>= ?q + Q)
}
```

The output of this phase is a proposed *Recovered Supply Plan* that minimizes recovery delays and costs. We identify a successful recovery as the case where all missing/reduced quantities from disrupted plans are provided alternatively. In this case, we insert the triple in the form *Plan isRecoveredBy xsd:string*, where we explicit which recovery strategy applied.

Resilience Evaluation Framework

In this section, we introduce step 4 in MARE, i.e., the evaluation framework for SC resilience and recovery. Thus, we compare the pre-disruption supply plans to the recovered supply generated in the recovery phase. We rely on the recovery performance evaluation metrics proposed by [47].

Recovery Cost Increase. is the extra expense caused by the disruption and the recovery as compared to the original price of the pre-disruption supply plans. First, we calculate the price of the recovered plan for each order, and we retrieve the order's original price. By summing the difference, we get the total cost increase for all orders in Listing 5.8. We do not consider the cost to rebuild anything physically lost as this is included in the severity factor.

Listing 5.8: Evaluate recovery cost increase.

```
PREFIX : <http://www.example.org/MARE#>.
SELECT (SUM(?currentprice - ?originalPrice) AS ?costIncrease) {
SELECT ?originalPrice (SUM(?price) AS ?currentprice
WHERE {
  ?order  :hasPlan  ?plan;
          :hasOriginalPrice ?originalPrice.
  <<?plan :needsPartner ?partner>> :hasQuantity ?q;
  :hasUnitPrice ?p; :hasTimeStamp ?t.
BIND (?p*q AS ?price)
} GROUP BY(?plan)
}
```

Recovery Speed. is the time taken till recovery is complete, i.e., for **S3**, it is the next available day where there is enough production capacity, entailing a new delivery time. In Listing 5.9, we calculate the number of orders where the delivery time in the supply plan is later than the original delivery time, pre-disruption. These orders are considered late orders, delayed by the difference between the original and the late delivery times.

Listing 5.9: Evaluate recovery speed

```
PREFIX : <http://www.example.org/MARE#>.
SELECT SUM(IF(?t>dt),1,0)) AS ?lateorders, SUM(IF(?t<=dt),1,0)) AS ?ontimeorders
SUM(?t-?dt) AS ?delay
WHERE {
  ?order :hasPlan  ?plan.
  ?order :hasDeliverDate ?dt.
 <<?plan :needsPartner ?partner>> :hasTimeStamp ?t
}
```

Unsuccessful Recovery. The ultimate goal of the SC is to deliver finished products to end customers, yet the result of the disruption caused by unplanned events can be unfulfilled orders as described by [194]. This metric is the count of the supply plans where all missing/reduced quantities from disrupted plans are not provided alternatively, i.e., *Plan isRecoveredBy xsd:string* does not exist. This situation occurs in case there is no alternative shipment mode or there is no strategic stock available, or there are no substitute suppliers to supply alternatively. Moreover, when we apply **S3: Delayed Recovery** if there was no free capacity within the next five days, we consider this as an unsuccessful recovery.

Customer Impact. The previous metrics can be calculated by SC stakeholders to analyze the impact of the disruption on specific customers. Within the customer relationship management paradigm, SC decision-makers apply recovery strategies in a way to attempt and reduce the impact of the disruption on high-priority customers.

5.2.4 Evaluation

In this section, we simulate the behavior of an exemplary SC under various disruptions scenarios and evaluate the SC recovery performance.

Experimental Setup

The following part presents the experimental setup for the proposed evaluation.

Supply Chain Structure. We consider a three-tier SC network consisting of one central node, i.e., an OEM (Original Equipment Manufacturer) directly linked to four suppliers in supplier tier 1 and four customers in customer tier 1, where C1 is the customer with the highest priority.

Supply Chain Data. We rely on the data generated and provided by the synthetic generator described in the technical report [8]. We simulate 400 orders and their corresponding supply plans, generated for a time-frame of 178 days, i.e., half a year.

Disruptions. We simulate the disruptions listed in Table 5.7. *:Disr1-4* have internal causes; accordingly, we apply **S1: Strategic Stock, S2: Alternative Shipment, S3: Delayed Recovery** consecutively. While *:Disr5* and *:Disr6* are external, i.e., affecting suppliers; thus, we apply **S4: Alternative Supplier**. Additionally, we create *:Disr7,8* that occur internally and externally; hence, we rely on a combination of the mentioned recovery strategies. Moreover, for conciseness, we show *hasDuration* which represents the length of the disruption in days, i.e. , *hasEndDate* minus *hasBeginDate*. The OEM in question relies on one transportation mode, so we cannot apply **S3: Alternative Shipment**.

Results

We propose a resilience evaluation framework as shown in Table 5.8 that incorporates the disruption characteristics, i.e., duration, severity, and the number of affected plans. Also, the framework includes the recovery metrics to evaluate the number of non-recovered plans, i.e., unsuccessful recovery, the percentage of total cost increase, and the delay. From the results in Table 5.8, we note that applying the strategic stock strategy leads to an increase in cost, whereas applying late recovery leads to delays in delivery. This impact varies based on the duration and the severity of the disruption as well as the number of affected plans. For instance, *:Disr2* has a duration of one day and a low severity affecting only two plans, thus the cost increase and the delays entailed are minimal. However, *:Dis1* and *:Disr3* have medium severity and a duration of three and five days, respectively. Therefore, the cost and delay are higher than in *:Disr2*. Likewise, *:Disr4* has a high severity and lasts for 45 days affecting 27 plans. Consequently, the entailed cost and delay are higher than the previously mentioned disruptions. Also, we note that for *:Disr5* and *:Disr6*, there is a significant cost increase since alternative suppliers can be more expensive than the firms' primary suppliers.

In case a disruption affects internally and externally *:Disr7* and *:Disr8*, there is a cost increase due to finding alternative suppliers and a delay in case of later recovery application. [47] explain that the longer it takes to fully recover, the more expensive the entire recovery process is likely to be. The

Disruption	Duration (days)	hasSeverity	affects Plan	Non-Recovered	Cost Increase %	Delay (days)	isRecoveredBy
:Disr1	5	Medium:0.3	13	5	0.053	0	Strategic Stock
				3	0	5	Late Recovery
:Disr2	1	Low:0.1	2	1	0.0054	0	Strategic Stock
				1	0	1	Late Recovery
:Disr3	3	Medium:0.3	12	3	0.026	0	Strategic Stock
				7	0	8	Late Recovery
:Disr4	45	High:0.7	27	20	0.17	0	Strategic Stock
				17	0	13	Late Recovery
:Disr5	100	High:0.7	124	0	0.32	0	Alternative Supplier
:Disr6	60	Medium:0.3	5	0	0.15	0	Alternative Supplier
:Disr7	4	Medium:0.3	16	11	0.08	0	Strategic Stock Alternative Supplier
				1	0.10	25	Late Recovery, Alternative Supplier
:Disr8	4	High:0.7	16	8	0.162	0	Strategic Stock, Alternative Supplier
				3	0.11	29	Late Recovery, Alternative Supplier

Table 5.8: Resilience evaluation framework to show the variation of the evaluation metrics values based on the different disruptions impacts.

delays caused by *:Disr8* are bigger than *:Disr7*. Thus, the cost increase is greater as with high severity disruptions and the consequences are severe.

In order for stakeholders to make more informed decisions, they can rely on the customer impact analysis as shown in Table 5.9 to examine the corresponding impact on specific customers. Consequently, they can decide which recovery strategy or combination of several to apply.

It is important that while applying recovery strategies, orders made by customers with high priorities whose plans are disrupted are recovered first. Therefore, we note that high-priority customers (C1) have fewer non-recovered plans than low-priority customers. Thus, their corresponding cost increase is higher than low-priority customers. Moreover, customers with low priority have longer delays because more important customers are recovered before; it might take more time periods to find the needed quantity to recover.

Discussion

MARE is a semantic model for DMP that enables SC visibility and data integration to simulate the performance of the SC under various disruptive events conditions. Nevertheless, MARE considers internal disruptions and external events that affect the supply. While external disruptions leading to sudden demand drops or surges can impact the SC badly if the SC is not equipped with suitable recovery strategies. Moreover, we address recovery performance in terms of SC metrics. It is essential

Disruption	affectsPlan				Non-Recovered				Cost Increase %				Delay			
	C1	C2	C3	C4	C1	C2	C3	C4	C1	C2	C3	C4	C1	C2	C3	C4
:Disr1	5	4	2	2	1	2	1	1	0.05	0.07	0.02	0.09	0	0	0	0
	5	4	2	2	0	0	1	2	0	0	0	0	2	2	1	0
:Disr2	1	1	0	0	0	1	0	0	0.01	0	0	0	0	0	0	0
	1	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0
:Disr3	3	3	3	3	0	0	1	2	0.07	0.21	0.55	0.3	0	0	0	0
	3	3	3	3	1	1	2	3	0	0	0	0	2	2	4	0
:Disr4	8	7	7	5	3	5	7	5	0.07	0.3	0.114	0.5	0	0	0	0
	8	7	7	5	1	4	7	5	0	0	0	0	7	6	0	0
:Disr5	12	24	37	51	0	0	0	0	0.6	2.44	4.65	6.99	0	0	0	0
:Disr6	1	1	1	2	0	0	0	0	0.04	0.1	0.18	0.37	0	0	0	0
:Disr7	4	4	4	4	1	3	3	4	0.015	0.03	0.04	0.41	0	0	0	0
	4	4	4	4	0	0	0	1	0	0	0	0.6	8	6	6	5
:Disr8	4	4	4	4	0	2	3	3	0.13	0.14	0.2	0.36	0	0	0	0
	4	4	4	4	0	0	1	2	0	0	0	0.6	6	6	6	11

Table 5.9: Customer impact evaluation with C1, customer with highest priority, having the least non-recovered plans and the smallest delay.

to extend to the decision chain and define who from the SC stakeholders is responsible and included in recovery as explained by [47].

5.3 Concluding Remarks

In this chapter, we presented KnowGraph-MDM and MARE, two semantic data integration approaches that support the SC become more efficient and resilient. KnowGraph-MDM enables an integrated reporting scheme for MD as well as the incremental development of the model via the intermittent stakeholders' involvement. With MARE, we proposed a semantic disruption management and resilience evaluation framework to integrate heterogeneous data sources covered by all DMP steps. KnowGraph-MDM and MARE rely on ontologies and KGs to create consistent and interoperable data exchange for SC applications. The output semantic models lead to a more enriched SC analysis to evaluate the behavior, and ultimately reach more efficient decision-making, especially needed during irregular circumstances.

CHAPTER 6

Semantic Web for Applied Semiconductor Supply Chain Management

We leverage SW technologies for applied SCM to address specific challenges entailed by the semiconductors domain. Via the contributions: [195], [196] and [197], this chapter answers the following research question

> RQ3: How can we apply semantic technologies to specifically support semiconductor supply chains?

We introduce KnowGraph-PM, a KG lead-time-based pricing approach allowing tailored revenue generation according to customers' profiles. Further, we propose SCIM-NN, a Semantic Context Information Modeling for Neural Networks based on ontologies, KG embeddings, and multi-stream neural networks to include context information for a classification task of COB. Also, we implement an ontology-based approach for preprocessing in machine learning, predicting the packaging required for a semiconductor product.

6.1 KnowGraph-PM: A Knowledge-Graph-based Pricing Model for Semiconductor Supply Chains

The bullwhip effect, a significant characteristic of the semiconductor SC, entails high inventory capacity, possibly unnecessarily. However, companies exploit revenue management ideas such as dynamic pricing to keep profitably utilizing capacity and encourage customers to have a more stable demand. Nevertheless, dynamic pricing strategies are myopic because the decisions are optimized, considering solely the expected profit obtained from the customer without any foresight on the long-run impact of these decisions [71]. Companies realize the need to maintain a good relationship with the customers while generating revenues.

Typically, interactions between revenue management and key account management have been largely ignored as both are often conducted independently of each other [198]. In addition, [199] explains how limited information sharing increases the difficulty of reducing the bullwhip effect and leads to inefficient SC management. Also, information integration increases the acquisition and maintenance of customers according to profitability [75]. Moreover, to the best of our knowledge, previously developed LTBP models solely rely on customer order behavior while not including the customer portfolio and specific characteristics. In this work, we create a KG that semantically integrates data from different sources, i.e., customers, orders, and customer account types. This solution suggests information integration from various data sources to customer-specific revenue management via LTBP. This improves revenue generation while maintaining customer relationships.

6.1.1 Implementation

This section describes our implemented approach, integrating data from customer relationship management and customer order data.

LTBP Knowledge-Graph (KG)

We use the ontology-based data access (OBDA) approach for semantic data integration. The schema is given in terms of an ontology representing the formal and conceptual view of the domain [94]. The data resides in the domain applications data sources.

Domain Ontology. Figure 6.1 shows the ontology comprising the *Order* class, which enables the coupling of the remaining classes of the domain, e.g., *Product* and *Customer*. The *Order* entity is uniquely identified via an *Order Number* and is described via twofold data properties. First, order information properties contain order descriptions such as the order quantity and the original price of the order. Second, an order is described with lead-time properties, e.g., order entry date, requested delivery date, and the standard delivery time. Through its two object properties *containsProduct* and *wasPlacedBy*, an order is linked to the *Product* and *Customer* entities respectively.

The *Customer* class describes a customer by assigning each customer a distinct customer code and by categorizing a customer into the defined customer account type or class. This categorization into a customer account type is then quantified with the data property *hasAdjustmentFactor*, which assigns each customer a specific pre-defined value based on their respective type. Finally, each product is uniquely identified with the help of the product number. Also, a *Product* includes data properties that contain all relevant product information such as the product basic type and the product line.

Data Sources. The use case described here belongs to Infineon. We utilize these two datasets: *DS1* that contains data about order details, e.g., lead-times, products, and corresponding customers. We filter orders containing products within one product line. We process roughly 65 thousand orders in the period between 2016-10-04 and 2020-09-01. *DS2* contains customer portfolio, i.e., customer account type, class, and location. It subsumes 177 customers and corresponding account types (regular, key, and others). Moreover, we use the corporate memory tool as a platform for data integration[1].

Lead-time-based Pricing Algorithm

The pricing algorithm is implemented as a SPARQL query that emulates the equation developed by [200]. SPARQL[2] is the query language standardized by the W3C for querying KGs. Figure 6.2 describes how parts of the equation match the nested query shown. Steps [1-4] depict how the

[1] lhttps://eccenca.com/products/enterprise-knowledge-graph-platform-corporate-memory
[2] https://www.w3.org/TR/rdf-sparql-query/

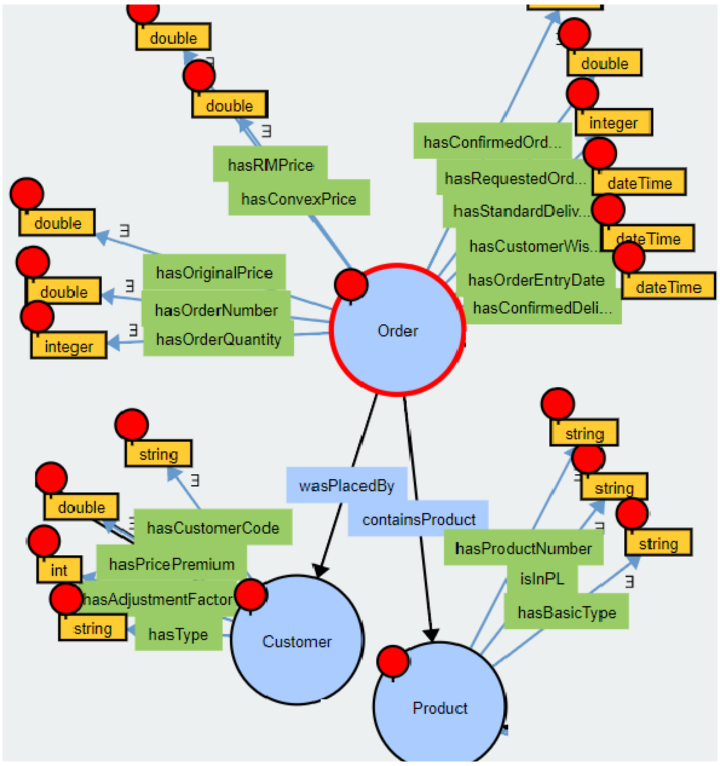


Figure 6.1: Domain knowledge-Graph comprising the order, customer and product classes with associated properties.

customer order behavior and customer account type contribute to calculating the price premium $P_{premium}$ for a specific customer. The last step is to calculate the new revenue management price P_{RMj} of a specific order j knowing the $P_{premium}$ of this customer.

1. We calculate the relative standard deviation (RSD) and the relative mean deviation (RMD) for each customer individually. This is done if revenue management is allowed, i.e., $SDT > OLT_{Requested}$, when customers request their orders earlier than *Standard Delivery Date*. This represents the customer order behavior from previous orders. α and β are coefficients to offer the model user the chance to emphasize one parameter more than the other. For instance, one could argue that the average deviation or mean is less important. For simplicity, alpha and beta are chosen to be equal to 1.

2. The sum of weighed RSD and RMD is then compared to a pre-determined threshold P_{max} which in this case is set to 2 as this represents the ceiling for the highest premium that can be offered to a customer which is double the original price. The sum is modified by the revenue adjustment parameter ρ. As shown in Table 6.1, for each customer account type or class, there exists a corresponding ρ based on revenue ranges. Large revenue ranges entail more important customer accounts having a greater advantage equivalent to higher ρ, leading to a smaller premium price.

3. In order to ensure that no customer receives a premium price smaller than the original we select

$$P_{premium_i} = Max\left\{\left\{Min\left[\left(\underbrace{\left(\sqrt{\frac{\sum_{j=1}^{n}(x_j-\mu_i)^2}{n_i-1}}\right)*\frac{\alpha}{\mu_i}}_{\text{RSD} * \alpha}+\underbrace{\left(\frac{\frac{1}{n_i}\sum_{j=1}^{n}(x_j-\mu_i)}{|\mu_i|}\right)*\beta}_{\text{RMD} * \beta}\right)_{①};(P_{max})\right]_{②}\right\}_{③}-\rho;1\right\}_{④}$$

```
INSERT { GRAPH <http://xyz.com/LTBP/> { ?customer :hasPricePremium ?premium }}
WHERE                                                                              ④
  { GRAPH <http://xyz.com/LTBP/> {
  SELECT
    ?customer (IF (?price < 1, 1, ?price) as ?premium)                             ③
  WHERE
      {
    SELECT ?customer (IF (?rsd + ?rmd >= 2, 2 -?ro, ?rsd + ?rmd - ?ro) as ?price) ?rmd ?rsd ?ro
    WHERE
      {                                                                            ②
      SELECT(AVG(?mean) as ?m)(AVG(?r) as ?rc)
      ?customer (COUNT(?order) as ?totalorders)
      ((math:sqrt(SUM((?x - ?mean) * (?x - ?mean)) / (?totalorders - 1)) / ?m) as ?rsd)   ①
      (xsd:float(SUM(xsd:float(?x - ?mean)) / abs(?m)) as ?rmd)
      WHERE
        { ?order :wasPlacedBy ?customer.
          ?order :hasStandardDeliveryTime ?sdt.
          ?order :hasRequestedOrderLeadTime ?rolt.
          ?customer :hasAdjustmentFactor ?r.
          BIND(?sdt*7 - ?rolt as ?x). FILTER(?x > 0 ).
          {
        SELECT ?customer (AVG(?x) as ?mean)
        WHERE
          { ?order :wasPlacedBy ?customer.
            ?order :hasStandardDeliveryTime ?sdt.
            ?order :hasRequestedOrderLeadTime ?rolt.
            ?customer :hasAdjustmentFactor ?r.
            BIND(?sdt*7 - ?rolt as ?x) FILTER(?x > 0). }
        GROUP BY ?customer } }
        GROUP BY ?customer
        }
    }
  }}
```

Figure 6.2: Lead-time-based pricing algorithm with the formula (top) and corresponding SPARQL query implementation (bottom).

the max between our first part of the formula and 1.

4. We assign $P_{premium}$ to each customer i where $1 < P_{premium} < 2$

5. Upon a new order j, we apply Equation 6.1 to get the revenue management price P_{RMj} from the original price P_{Oj} of the order and customer-specific $P_{premium}$ assigned in the previous step. As mentioned, the difference between SDT and $OLT_{Confirmed}$, is exploited for revenue management.

$$P_{RMj} = P_{Oj} + P_{Oj} \times [(1 - \frac{OLT_{Confirmed_j}}{SDTj}) \times (P_{premium} - 1)] \tag{6.1}$$

Revenue (r) Range	Customer Class	Price Adjustment Factor ρ
10<r<100m	Key	0.1
5<r<10m	Regular	0.05
0<r<5m	Others	0.025

Table 6.1: Exemplary values of price adjustment factor varying per customer class and revenue range.

6.1.2 Evaluation

The implemented approach combines data from customer relationship management with customer order behavior. We evaluate KnowGraph-PM twofold. First, we check if the created KG covers the domain in question. Second, we calculate the total revenue generated using static and dynamic pricing algorithms and compare them with our KnowGraph-PM model.

Competency Questions and KG Evaluation

We translate the listed competency questions, defined by domain experts, to SPARQL queries shown in Appendix E. We execute them on the described KG in Section 6.1.1. The results are extracted and represented in Figure 6.3.

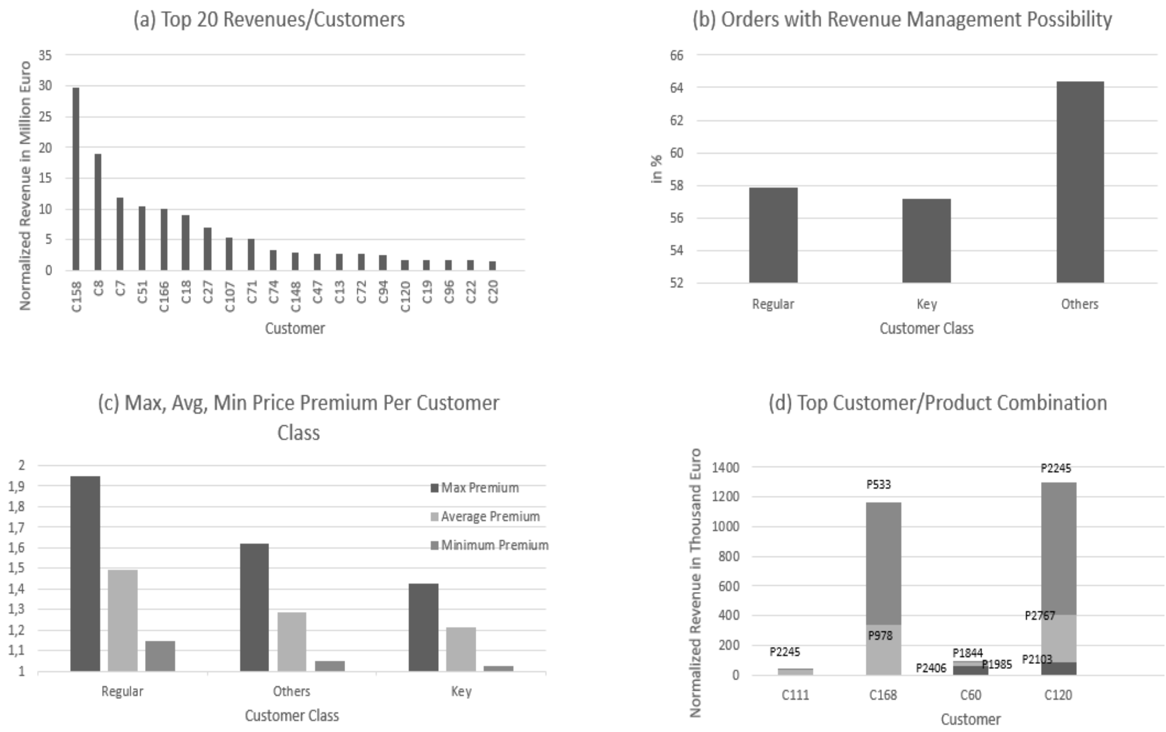


Figure 6.3: Chart visualization of SPARQL query results corresponding to different competency questions.

CQ1: Get top 20 most profitable customers: This competency question retrieves the top 20 customers who generate the highest revenues when they buy the Order with the assigned price premium. The results shown in Figure 6.3(a) give a valuable insight into the most influential customers for a firm. This can be extended by examining these customer classes and accordingly tailoring marketing strategies.

CQ2: Retrieve LTBP occurrence-based customer ranking**:** We can examine the overall customer order behavior of the different customer classes. For this, we rank customer types based on the highest percentage of occurrences of $SDT > OLT_{Requested}$ (RM is possible). We analyze which customer classes are more likely to request a delivery date that is earlier than the *Standard Delivery Date*. This makes them eligible for revenue management with the adjustment factor ρ in Table 6.1 suitable to their respective customer class. The results in Figure 6.3(b) show that *Key* account customers are least likely to request an early delivery. This is an expected outcome. Key customers have strong relationships established with the company. Thus, they are more likely to enter long-term agreements and contracts entailing stable order behavior, i.e., limited variation in lead-times. Additionally, because of the frequency of their past orders, they can accurately forecast their needs and how a firm can fulfill them. Accordingly, this constrains dynamic pricing and revenue management potential.

CQ3: Calculate per customer class price premium estimation: This question provides information about the customer class that is most likely to be the most profitable upon applying revenue management. The information contains the highest, the lowest, and the average Price Premium paid by customers of each customer class. Also, it gives insights into the customer order behavior of the different customer classes. We observe in the results shown in Figure 6.3(c) that the Regular customer type has the highest maximum, average, and minimum price premiums. This means that this customer class would yield the highest average revenue when the dynamic pricing model is applied.

CQ4: Select initial customer and product for LTBP: This CQ determines the combination of customers and products that a company releases with the first practical implementation of the LTBP model in order to maximize the initial potential revenue increase using the adjusted prices. Results in Figure 6.3(d) provide the combination of customers and products in terms of profitability and revenue generation in the case where the lead-time-based pricing is applied.

Revenue Management (RM)

Dynamic pricing as part of RM is about generating revenue for the company. Existing dynamic pricing algorithms, e.g., the convex model, consider customer behavior while ignoring customer profiles. KnowGraph-PM customizes the RM price based on customer order behavior and customer relationship with the company. The results show that $TotalOrginalPrice < TotalRMPrice < TotalConvexPrice$. In that, *TotalOrginalPrice* is for the case that no RM is applied, which indeed is the summation of original prices for each order P_{Oj} as provided in Equation 6.1, *TotalRMPrice* is the sum of RM price, i.e., P_{RMj} as of Equation 6.1, and *TotalConvexPrice* is the sum of order prices if Equation 2.1 is applied.

These results indicate that by applying KnowGraph-PM, a company can generate revenue while considering the customer portfolio. Other models, such as the convex, generate higher revenue but can entail a disturbed relationship with the customer. The implemented approach integrates data from customer relationship management as well as customer order data to generate a price premium that fits the customer portfolio. The proposed solution is portable and can be applied to other domains with the right mapping between data and ontology. Additionally, the two evaluation techniques show that it is up to the company to adapt pricing strategies based on customer portfolio. SPARQL queries can be tweaked to show details about a specific customer, consequently tailoring marketing strategies to fit different customers.

Listing 6.1: Evaluation query to calculate the revenue management price and the original price.

```
PREFIX : <http://www.example.org/DynamicPrice#>.
SELECT (sum(?rmprice) as ?TotalRMPrice) (sum(?orignalprice) as ?TotalOrginalPrice)
  (sum(?convexprice) as ?TotalConvexPrice)
FROM <http://xyz.com/LTBP/>
WHERE {
  ?order :hasRMPrice       ?rmprice.
  ?order :hasOriginalPrice ?orignalprice.
  ?order :hasConvexPrice   ?convexprice.
}
```

Discussion

We identify a limitation in the evaluation of KnowGraph-PM as we compare our model to a convex model with α chosen to be -0.5 in Equation 2.1. This choice is limited as the parameter is set after using simulation models to optimize the revenue for a specific set of customers. Second, for the evaluation, we calculate the total revenue assuming that all customers accept the price change. However, in practice, customers can refute the premium price and stick to the original. In future work, we aim to integrate into KnowGraph-PM an acceptance rate distribution to model customers' behavior toward dynamic pricing. This would lead to more realistic figures in the evaluation, thus, overcoming the mentioned limitation.

6.2 SCIM-NN: Semantic Context Information Modeling for Neural Networks in Customer Order Behavior Classification

Understanding customer order behavior helps the semiconductor SCs achieve more accurate demand forecasts. Classifying COB in patterns using a ML algorithm improves grasping customer behavior. Furnishing context and elaborate descriptions of the domain to the ML algorithm impacts the performance of the classification task. Semantic models capture domain knowledge in a comprehensive way that provides high level description of the domain. Thus, we present a methodology to semantically model customer context information for a neural network classifying customer order behavior. This work is a joint collaboration with the co-authors. Our part in this contribution focuses on building the first streamline to collect the relevant context information and semantically conceptualize it and deploy it into the two streamline CNN.

6.2.1 Methodology

Figure 6.4 depicts the Semantic Context Information Modeling for Neural Networks (SCIM-NN) methodology used to include semantic context information into a neural network classification approach. Two streams are used separately and are merged for a final context-aware classification.

- *Context Stream:* Handles context information in the KG.
- *Neural Network Stream:* Handles the main classification task which is augmented by context information.

The upcoming sections explain both streams and the final KG-based context-aware model in more detail.

Context Stream

The *Context Stream* gathers, integrates, and processes context information chosen in an evaluation performed by domain experts. KG embeddings generate vector representations for the consolidation with the *Neural Network Stream*. The primary stages of *Context Acquisition*, *Integration*, and *Utilization*, as seen in Figure 6.4 are described further.

Context Acquisition. The first step in the *Context Stream* includes a *Context Source Connector* that is capable of acquiring context information from a context source manually, semi-automatically, or automatically. The inputs that a context source connector can handle and the degree of automation are implementation-dependent. An example is a context connector getting context information through an API from general-purpose KGs like DBpedia or Wikidata.

Since the quality of context information varies, preprocessing steps such as *Cleaning and Transformation* steps are integral to ensure a high-quality KG. Such steps include handling missing information or identifying and correcting outliers manually or automatically with statistical approaches.

Context Integration. After gathering and cleaning the context information, we create a KG that contains the context information and relationships. An ontology specifies classes as well as properties, which are used in the KG in the *Ontology Mapping* step. The ontology is essential for the approach since it allows to use the Semantic Web technologies such as reasoning to validate the context information.

Figure 6.5 shows a section of the context ontology used for *Ontology Mapping*. The ontology partially reuses classes and properties from standardized and well-known ontologies such as the W3C Organization [160] and the W3C Time [161] ontology. The *Context* class is used for individuals representing generic context types such as customer forecasts or customer industry. Actual context information in numerical form or other formats is modeled by the *Context_Information* class. An example would be a customer forecast of 1000 as an individual of class *Context_Information*. However, we model the temporal aspect of context information on a different granularity to express to which point in time context information is associated to. *Context_Node* is modeled as a *Temporal_Relation*

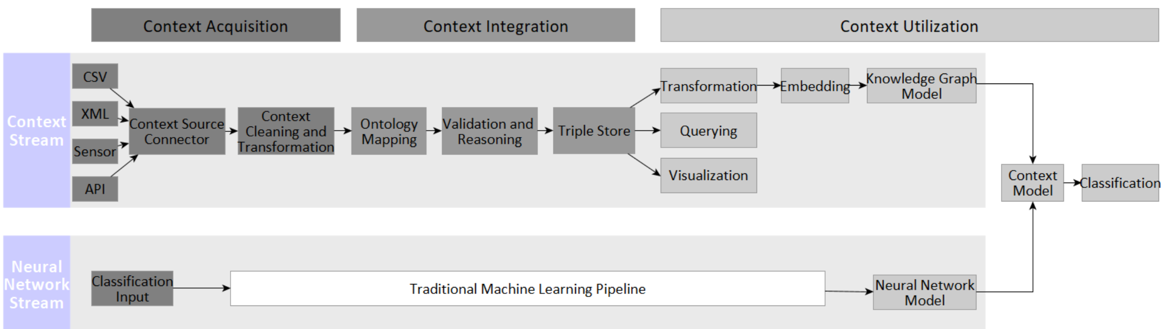


Figure 6.4: Semantic Context Information Modeling for Neural Networks (SCIM-NN) methodology.

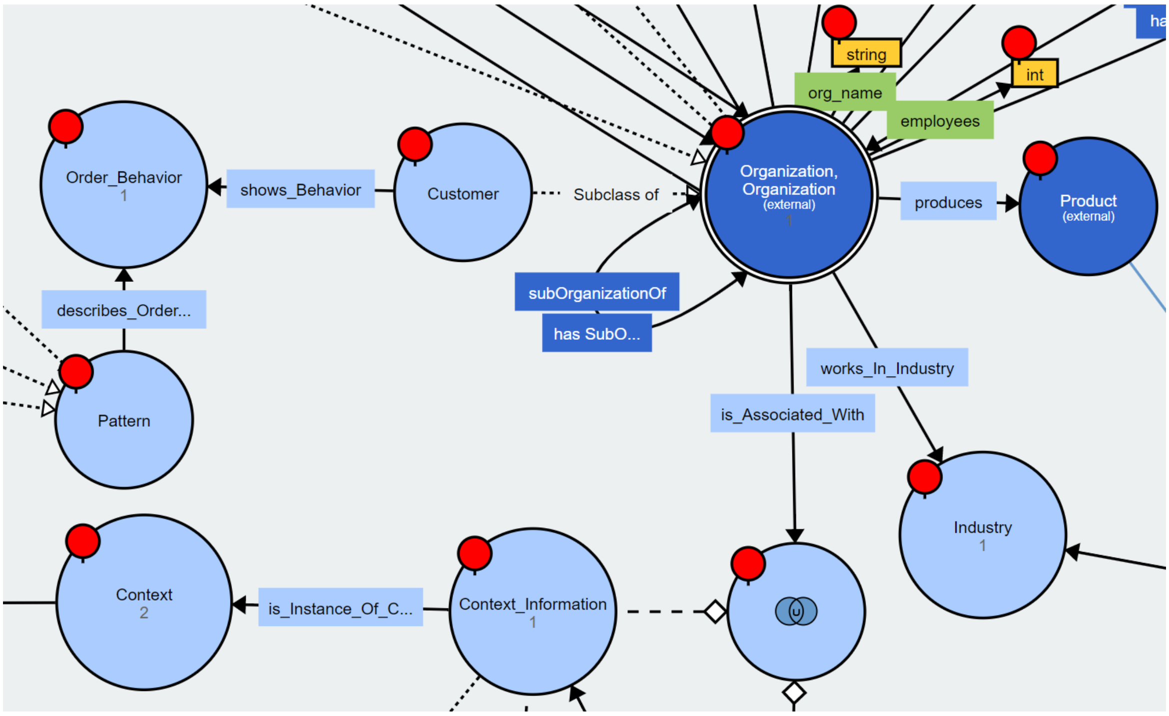


Figure 6.5: Part of the context ontology with the major classes context and context information.

according to the N-ary relations pattern [201]. This allows the aggregation of context information on different temporal granularity.

Then, we map the context information to the classes in the created ontology and create the context KG. *Reasoning* validates the context information and ensures a high-quality KG based on the constraints defined in the ontology. For example, each customer can only have one associated name but might have multiple industries. In case of an inconsistency detected by the reasoner, an error is thrown, and context information needs to be adapted accordingly. Afterward, we store the context KG in a triple store for accessibility and subsequent steps.

Context Utilization. At this point, the context information in the created KG is stored in *Triple Stores*. A triple store is a database for the storage and retrieval of triples. The KG can be utilized through various means such as *Querying*, *Visualization* of the ontology and semantic data, or by using it for machine learning.

The triples from the KG are then transformed into the required input format in the *Transformation* step. This step is needed since embedding libraries require specific data input formats, e.g., comma-separated values (CSV) of the triples. Furthermore, a KG embedding model generates a dense numerical representation of the entities from the graph, capturing literal information as well as semantic relationships in the *Embedding* step. For each context information, a separate embedding is generated from the graph.

$$v_1 = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} v_2 = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} v_m = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix} \tag{6.2}$$

For each context vector v, based on the amount of different context information used for the approach, m context embedding vectors are generated. Each vector v has a length of n which is determined by the used KG embedding model and the dimensionality hyperparameter. The parameters x, y and z represent embedded context information from the KG captured by the embeddings. Equation 6.2 defines how embedding vectors for context are created.

The resulting concatenated embeddings are the input to the *Knowledge-Graph Model*, which is a neural network with several dense and dropout layers to capture relevant features from the high-dimensional context embeddings. The number of dense and dropout layers, layer sizes, and overall model architecture depends on the use case and the complexion of context information. The model learns context features associated with the primary classification task, which are later merged within the *Context Model*.

Neural Network Stream

The *Neural Network Stream* handles the main classification task with corresponding input data that should be augmented by the context information from the *Context Stream*. It can be based on a fully connected neural network architecture, a convolutional neural network (CNN), or other architectures. *Traditional Machine Learning Pipeline* refers to steps that are usually done to prepare a data set for a classification task. This contains, for example, preprocessing or data enrichment steps like sampling. [79] explains such a pipeline for the COB use case. A *Neural Network Model* as a result handles the main classification task in the stream.

Context Model

In the end, the *Context Model* uses the concatenated outputs from both the *Context*- and *Neural Network Stream*. Here, a dense layer with subsequent optional dropout is applied to jointly learn features on the neural network and context information after the fusion operation. Finally, a softmax layer generates the class probabilities for a context-aware classification incorporating the KG context.

6.2.2 Application on Customer Order Behavior

In this section, we apply SCIM-NN to a use case for COB. Five customers of Infineon are considered for the implementation of the *Context Model*. We include heat maps detailing the customers' order behavior as well as associated context information on the granularity of delivery weeks.

Context Information Data Set

Firstly, we perform an evaluation of relevant context information for the use case. We conduct interviews and discussions with domain experts in the area of Customer Logistics Management and COB at Infineon to identify relevant context information.

Afterward, an ontology-based approach specifies classification properties for context information with scores defined by the domain experts. Finally, we use SPARQL to aggregate an average score for each context information. The decision on which context information to include in the KG is based on the final scores.

Based on the sorted scores, the following context information is included for the use case:

- **Semiconductor Index** as semiconductor market-related context information. Semiconductor index information, and financial information are retrieved from the finance portal Yahoo finance.
- **Customer forecasts** as order-related context information. Customer forecasts are numerical forecast values from which the heat maps are generated. These are retrieved from Infineon databases.
- **Organization information** such as industry, employees and financial information like assets and revenue as company-related context information. It is fetched from Wikidata [202], a well-known general-purpose KG and financial information from the Bloomberg Terminal. Wikidata enables the reuse of already present semantic data for the KG of the approach and simplifies an extension to include further customers.

Context information like Semiconductor Index and forecasts, which dynamically change, are included in the granularity of the delivery week. On the contrary, information like the customer's industry does not change frequently and stays the same across delivery weeks. Revenue, employees, and assets are included as static context information representing the current customer situation.

Heat Map Data Set

Heat maps for all customers are included on a delivery week basis, similar to the context information. [79] uses synthetic heat maps for the training of the CNN model and shows that a combination of synthetic and real data improves the performance of the model. However, only real customer heat maps can be included for the context-aware approach. Synthetic data generation would be feasible if the underlying structure of the real data is well understood and can be used for the generation of synthetic data. In practice, the complex relationships between context information and COB patterns are not easily conceivable, which hinders the creation of synthetic heat maps. Additionally, synthetic context information representing markets and synthetic customers with employees and industries are not easily reachable. Therefore, we include only real customer heat maps in the approach. These heat maps need to be labeled and can contain more complex patterns and missing values compared to synthetic data.

Since only real customer heat maps can be used, the distribution of class occurrences is heavily skewed towards the majority class Random. As synthetic data generation is not an option for balancing the training data set, we rely on sampling methods. We apply oversampling with a factor of 4 on the minority classes and undersampling with a factor of 0.7. The factors result from our data structure and the identified imbalance of classes. The outcome is a more balanced training set for the *Context Model*. Figure 6.6 shows the COB class distribution of the labeled training heat maps after sampling has been applied. Random and Up-Down are the most common classes with roughly 25%. Classes, which rarely occur in the real customer heat maps, are Down-Up, Up-Down-Up, and Double Booking, with a share of 1.1% to 5.6%.

Customer Context Stream			**Heat Map Stream**		
Layer	Channel	Activation	Layer	Channel	Activation
Input	300	ReLU		Refer to [79]	
Dense	24	ReLU			
Dense	24	ReLU			
Dropout					
Context Model					
Layer		Channel		Activation	
Concatenate					
Dense		256		ReLU	
Dropout					
Dense				Softmax	

Table 6.2: Architecture of the implemented customer order context model based on SCIM-NN.

Customer Order Context Model

The *Customer Order Context Model* uses the heat maps and context information for context-aware heat map classification. It contains the *Customer Context-* and *Heat Map Stream*. Table 6.2 depicts the architecture of the *Customer Order Context Model*.

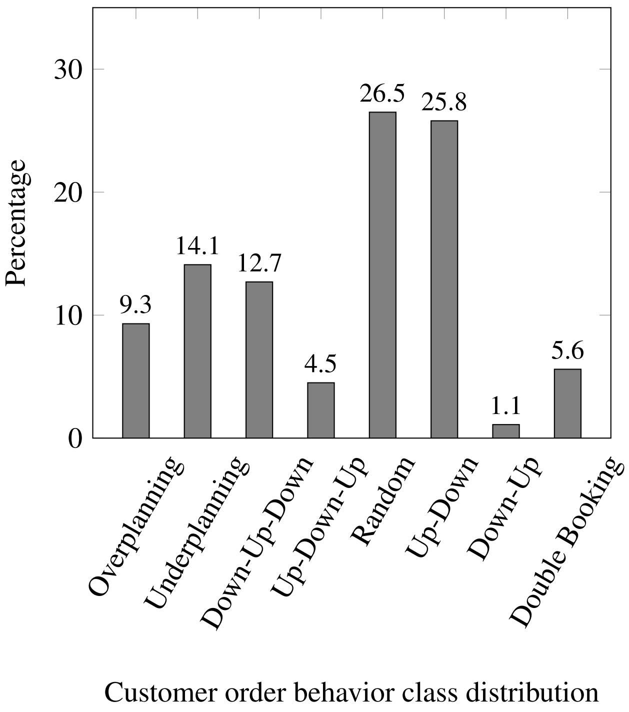


Figure 6.6: Labeled training data per order behavior class.

Customer Context Stream. The *Customer Context Stream* gets the before-mentioned context information as input and generates a KG based on the ontology.

Included context information can be further utilized by querying with SPARQL. The KG-based approach aggregates context information by averaging or other analytical queries. Listing 6.2 depicts a SPARQL query to aggregate and average the first month of Semiconductor Index weekly values in the business year.

Listing 6.2: SPARQL query to average Semiconductor Index information for the first business month of 2022.

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX time: <http://www.w3.org/2006/time#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX cxt: <http://www.example.org#customer-context#>
SELECT ?year (AVG(?semIdxVal) AS ?avg)
WHERE {
  ?semIdx rdfs:label "Semiconductor␣Index";
          cxt:Semiconductor_Index ?semIdxVal;
          cxt:has_Time ?temporalDescription.
  ?temporalDescription time:week ?week;
                       time:year ?year.
  FILTER(?week >= 1 && ?week <= 4 &&
         ?year = 2022)
}
GROUP BY ?year
```

For the embedding of context information from the KG, LiteralE in conjunction with ComplEx is used [145]. Firstly, LiteralE can embed numerical information like revenue or customer forecasts. Secondly, LiteralE with the baseline model ComplEx can produce a highly competitive performance compared to other KG embeddings on various benchmark data sets [203]. The embedding step generates three context embedding vectors for the Semiconductor Index, customer forecasts, and organization context information. ComplEx generates 100-dimensional embeddings, which leads to an input layer of size 300 for the neural network in the *Customer Context Stream*. Several dense layers with a consecutive dropout layer, as seen in Table 6.2, are used to learn complex features from the context embeddings. We rely on a rectified linear activation function or ReLU for input and dense layers.

Heat Map Stream. A CNN developed by [79] pretrained on synthetic and real heat maps is used for the main classification task in the *Heat Map Stream*. Therefore, classification accuracy on the heat maps is already high, with an accuracy of 69.2% on average for fifty different data set test splits. The accuracy is lower compared to [79] based on two differences in the models. First, the classes Constant and Ten Times Booking, which show a very high accuracy, are not present in our data set based on the granularity. Second, [79] uses a greater amount of synthetic data for the training of the model. The last softmax layer of the CNN is removed to allow for further concatenation with the *Customer Context Stream*.

Context Model. The *Context Model* gets the output of both the *Customer Context Stream* and *Heat Map Stream* to fuse it into one neural network for the final classification. Furthermore, a dense ReLU layer with the following dropout layer is applied jointly to learn features on context information and heat maps. Finally, a softmax layer generates the class probabilities and context-aware classifications for the heat maps.

6.2.3 Evaluation

We outline in the following section the used resources and the experimental design for the implementation of the described *Context Model* on the COB. Afterward, results of the *Context Model* compared to a benchmark model are presented with a subsequent discussion of the benefits and limitations of the approach.

Experimental Design

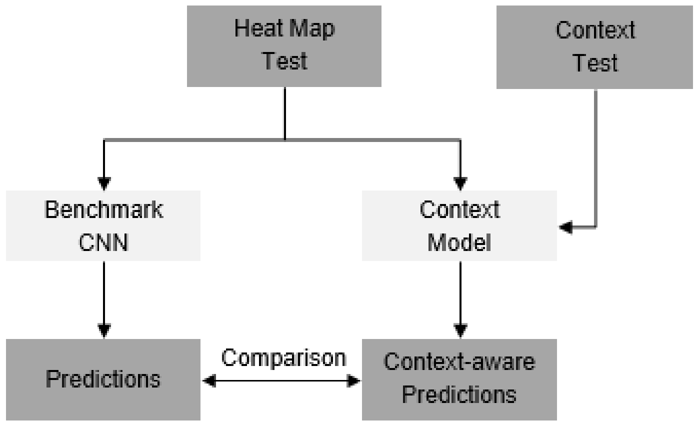


Figure 6.7: Evaluation approach and comparison with benchmark model.

We use Infineon's compute farm to train the *Context Model* on a 16 core CPU with 16 GB of RAM. The training data set consists of 1,411 heat maps after sampling is applied. Additionally, 171 heat maps are kept for each validation and test set. Sampling is applied after the split into training, validation, and test set to avoid samples from the training set being present in the validation or test set. This is the input to the *Heat Map Stream*. Additionally, the same amount of training, validation, and test data are used as the input for the *Customer Context Stream*.

The workflow implementation of the methodology is implemented with Python. Context information is extracted from CSV files. Data cleaning steps are unnecessary since all context information is extracted from structured data sources that exhibit high quality. Ontology modeling is done manually, while the creation of the KG from context information, as well as ontology mapping, is automated with Owlready2 [204]. Subsequent reasoning for the validation of the KG is done with HermiT [165]. An XSLT stylesheet transforms the triples from the KG into the required input format for the embedding library. The LiteralE KG embedding model creating the embeddings uses the embedding library PyKeen [205].

We use the CNN developed by [79] as a benchmark for the developed *Context Model*. This model is also used for the implementation of the *Heat Map Stream* in the *Context Model*. However, the layers are frozen to ensure better compatibility of both models. Therefore, the CNN is not further trained on the customer heat maps. Only the last layers of the *Context Model* and the layers in the *Customer Context Stream* are trained as seen in Table 6.2.

For the evaluation, as seen in Figure 6.7 both the *Context Model* and benchmark CNN model get the same heat maps as the test set. Additionally, the *Context Model* receives the matching context information as an input. Fifty seeds are used to generate 50 different data set splits and associated models to mitigate the small data set size. The average performance of both models is then compared in Section 6.2.3.

Results

The *Context Model* is compared to the benchmark model explained in Section 6.2.2 based on different performance metrics. Table 6.3 depicts the average accuracy for both the *Context-* and benchmark model.

Including context information resulted in a 1.24% higher accuracy in the classification of heat maps. Furthermore, an individual per-class performance comparison for both models was conducted to get further insights into the performance of the important minority classes. Table 6.4 shows the averaged performance of both models for all classes based on a comparison of the true and classified labels. The results show that the better accuracy of the *Context Model* is attributed to the better performance on the majority class Random. Based on the results, an impact of the context information from the KG on the classification is visible. Overall, the *Context Model* performed better than the benchmark model for five and worse for three classes.

On average, the *Context Model* performed by 43.7% better on Double Booking and 15% better on Up-Down-Up. These classes show the most significant difference in the classification performance in favor of the *Context Model*. Random, Down-Up-Down, and Underplanning show a minor improvement of about 2% compared to the benchmark model. This improvement is beneficial since Random and Underplanning are commonly found patterns in the customer heat maps and add more business value when classified correctly. An explanation of the improved performance on Underplanning and Up-Down-Up might be the inclusion of the customer itself as context information. Specifically, a few customers showed those patterns and, therefore, an organization as context could help predict these classes. However, more research on the impact of context information needs to be conducted. On the flip side, Down-Up, Overplanning, and Up-Down show a worse performance compared to the benchmark. While the worse performance on Down-Up might be attributed to a small training set, Up-Down and Overplanning need to be investigated further since the results are based on a more than eight times more extensive training set than Down-Up.

Results show a better accuracy of the Context Model overall and better performance on five out of the eight COB classes. Therefore, adding context information using semantic models shows a positive effect on the model. The decreased performance in some classes is due to insufficient real customer

	Accuracy
Context Model	70,50%
Benchmark Model	69,26%
Difference	**+ 1.24%**

Table 6.3: Overall accuracy of the context model is 1.24% higher than the benchmark model.

	Double Booking	**Down-Up**	**Down-Up-Down**	**Over planning**	**Random**	**Under planning**	**Up-Down**	**Up-Down-Up**
Context Model	**86.00%**	2.00%	**23.20%**	73.40%	**73.58%**	**91.00%**	70.24%	**40.32%**
Benchmark Model	42.30%	**40.00%**	21.20%	**85.80%**	71.70%	89.20%	**75.00%**	25.30%
Difference	**+ 43.70%**	**- 38.00%**	**+ 2.00%**	**- 12.36%**	**+ 1.90%**	**0,0.502,0**+ 1.80%	**- 4.80%**	**+ 15.00%**

Table 6.4: Evaluation of per class performance of the Context Model compared to the benchmark model.

data and associated context. Classes like Down-Up, Double Booking, or Up-Down-Up, which rarely appear in real customer data. More training data allows the model to learn better possible context patterns related to all order behavior classes.

Discussion

An advantage of SCIM-NN is the ability to leverage Semantic Web and KG tools to utilize context information in a machine learning environment. Reasoning ensures adherence of the context information in the KG according to restrictions and limitations given by the ontology. SPARQL can be used to query, aggregate, or update the context information. Additionally, triple stores used to store the KGs contain useful features for the visualization of the graph. Furthermore, we can easily extend the KG with additional context without changing the underlying ontology. Lastly, the ontology as the domain model represents a human-understandable documentation that helps communicate complex relationships in the domain.

These further context utilization options can help elevate KGs from a machine learning input to a mature data structure surrounded by a stack of SWT and tools. A drawback of the proposed approach is the need to train multiple models (KG embedding model and neural networks) for the final context-aware model. This need for extensive training can be expensive in terms of time and effort.

6.3 An Ontology-based Approach for Preprocessing in Machine Learning for Packing Information

In dispersed SCs, complex data flows potentially entail missing or wrong data values. In an attempt to compensate for the missing values, ML algorithms predict the target values, which helps preventing the users and experts from adding manually incorrect values to the system and causing operational complications. In fact, the performance of the prediction algorithm relies heavily on the quality of the input data. We explore the use of SWT to clean and understand the input data in a preprocessing phase for a ML algorithm.

In this use case, we examine a ML model predicting the packing information for semiconductor products. We refer to the packing info as the data object that includes information on how a product is stored and shipped to the customer. It contains a list of packing materials used for packing a finished product and additional packing information like packing weight and outer box dimensions. During the preliminary phases for this use case, we conduct a prior data analysis where experts establish that several packing info data quality is insufficient. The major reasons affecting the quality of the packing data are the missing and incorrect values for the outer box dimensions (width, length, and height) and total packing weight. As emphasized by the domain experts, having correct and complete values is important for customs declaration in deliveries to various countries and to avoid triggering errors in the system during the Distribution Center (DC) automation activities.

We rely on a ML algorithm to predict the missing target packing values. In order to increase the quality of the input data for the algorithm, we propose a framework that leverages SW artifacts, e.g., ontologies, SPARQL, and reasoning, to detect and correct insufficiencies in the packing info data element. This work is a collaboration with the co-authors. Our contribution is the ontology modeling as well as the constraints design and SPARQL queries writing to detect outliers and missing values. Also, our contribution extends to the overall system architecture design and the deployment of the semantic models into the system.

6.3.1 Framework

In this first part of the system architecture, we focus on improving the input data quality. As depicted in Figure 6.8, based on an input file containing the packing info, the system creates a domain ontology to enable data understanding and the use of reasoning and queries to address the data cleaning preparation. The output of this phase is a clean data report used for the further preparation steps, namely, the selection and transformation of the features. After the preprocessing phase, the system implements the ML model to predict the total packing weight and outer box dimensions (width, length, and height).

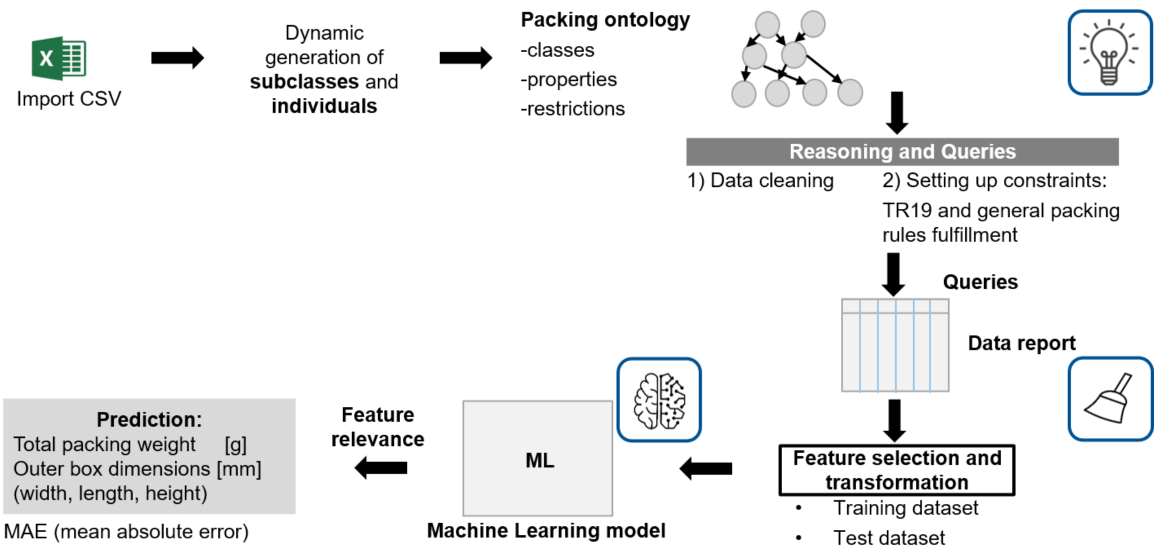


Figure 6.8: System overview of semantic-based preprocessing for packing information prediction.

Assumptions

We propose the semantic-based preprocessing framework based on two fundamental assumptions. First, we derive the packing knowledge for the elaboration of the domain ontology from the internal documentation of our use case. Likewise, the knowledge needed to elaborate the competency questions is sufficient based on the experts' opinions and is therefore case-specific. Second, this contribution focuses on exploring the influence of a preprocessing stage implemented utilizing an ontology and other SWT, on the performance of a ML model. With this goal in mind, the methodology analyzes the performance of the model with its default parameters without considering further hyperparameter optimization of the model.

Data Understanding

We create a packing ontology to model the domain following the ontology engineering process by [175]. Figure 6.9 shows the basic classes of the packing ontology and their connecting properties. The ontology's goal is to answer the competency questions in Table 6.5 that reflect an appropriate (use case-specific) packing domain coverage. **CQ1** retrieves all object and datatype properties of the packing info, an instance of the class *packing*. The goal of **CQ2** is to get the relation between a sales product (a subClass of *semiconductor_product*), a package (via *has_package*) and a packing info as an instance of a packing, which is related to class *package* via *is_assigned_to* property. **CQ3** selects functional packing methods, a subclass of *dice_packing*, sublcass of the class *packing*. **CQ4** similar to **CQ2** established the relation between the *semiconductor_product* and packing info. Then, we introduce the relation *included_in* between *packing* and the *packing_material*. **CQ5** extends **CQ2** to retrieve the datatype properties and their values packing info.

Data Preparation Process

The data preparation process consists of **Data Cleaning Preparation** and **Feature Transformation and Selection**. The data cleaning preparation phase includes inconsistency and outliers detection and correction as shown in Table 3.2.

Data Cleaning Preparation. In this phase, we identify two types of inconsistencies: missing values and outliers.

For **the detection and handling of missing values**, we rely on reasoning and querying. The use of the reasoner alerts the presence of a datatype different from the datatypes stipulated in the restrictions. For instance, during the creation of the ontology, the missing values are replaced with "nan" which stands for "not a number". Hence, the presence of this string datatype is alerted by the reasoner as inconsistent. Moreover, we leverage the SPARQL query in Listing 6.3 for the identification and elimination of the triples with missing values in order to ensure completeness in the data represented in the ontology. This query searches for the class members whose object value is equal to "nan", and once the triple patterns match, the DELETE clause removes them.

No.	Question
CQ1	What are the properties of a packing info?
CQ2	How is a packing info assigned to a product?
CQ3	How is a functional packing assigned to each product type?
CQ4	What are the materials used for packing a product?
CQ5	Which values of packing weight and outer box dimensions does a product have?

Table 6.5: Competency questions to evaluate the packing ontology model to find the packing knowledge necessary for the user to implement the ML model.

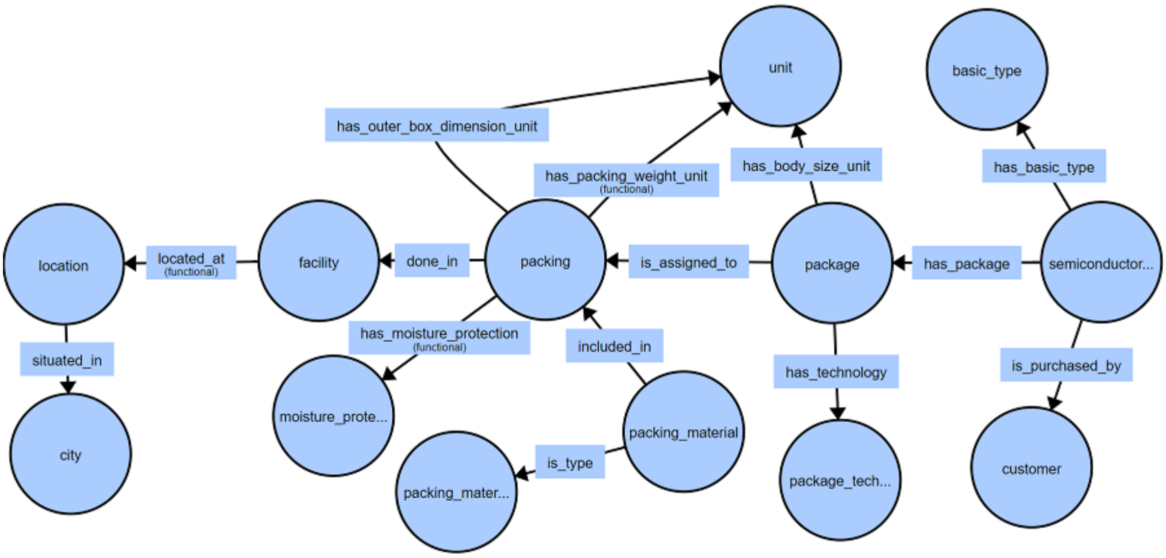


Figure 6.9: Main classes in the packing ontology: Product, packing, package and facility.

Listing 6.3: Handling missing values.

```
DELETE { ?instances ?predicate ?object }
WHERE {
  ?instances rdf:type :class.
  ?instances ?predicate ?object.
FILTER (?object= "nan")
}
```

Likewise, it is possible to distinguish the classes missing a parent class and the instances missing a relation to other classes. The example query in Listing 6.4 retrieves the classes which are not members of any of the parent classes (e.g., small_reel and big_reel) and removes their direct relation to the superclass (e.g., reel). Then, based on the domain knowledge, the result will be the correct classification of the subclasses to the suitable size class (e.g., big_reel or small_reel).

Listing 6.4: Adding classes to unclassified subclasses.

```
DELETE { ?reel_subclasses rdfs:subClassOf :reel}
INSERT { ?reel_subclasses rdfs:subClassOf :big_reel}
WHERE {
  ?reel_subclasses rdfs:subClassOf :reel.
  ?o owl:hasValue ?value.
  ?o owl:onProperty ?property.
  ?reel_subclasses ?p ?o.
FILTER NOT EXISTS{ ?reel_subclasses rdfs:subClassOf :small_reel.}
FILTER NOT EXISTS{ ?reel_subclasses rdfs:subClassOf :big_reel.}
FILTER(?property= :has_outer_diameter && ?value=330.0)
}
```

For **the detection and correction of outliers**, we split the outliers into error outliers, and single construct outliers [206]. We define the first as the differing values within the data due to the inaccuracies or errors in sampling, computing, preparing, or data manipulating. The second outliers are values that are unusually large or small in comparison to the rest of the data. While the latter type of outliers is possible to identify in the tails of the data distribution, the error outliers are harder to detect.

The reasoner identifies the inconsistent subclasses and instances that belong to multiple disjoint superclasses. Then, a SPARQL query, as in Listing 6.5, detects the errors identified by the reasoner, eliminates the incorrect triple, and keeps the triple that represents the correct superclass, according to the specific property and value. At the end of this example, the subclasses will only belong to *big_reel*.

Listing 6.5: Eliminating an incorrect triple based on a property and a value.

```
DELETE { ?reel_subclasses ?p ?o_original}
WHERE {
  ?reel_subclasses rdfs:subClassOf :small_reel.
Filter exists { ?reel_subclasses rdfs:subClassOf+ :big_reel.}
  ?reel_subclasses ?p ?o_original.
  ?reel_subclasses ?p ?o.
  ?o owl:hasValue ?value.
  ?o owl:onProperty ?property.
FILTER (?o_original= :small_reel && ?property= :has_outer_diameter && ?
    value=330.0)
}
```

Moreover, the use of queries supports the data exploration and constraints construction to detect and correct error outliers. Query in Listing 6.6 corrects the has_outer_diameter property for small_reel by replacing the errors with the right values.

Listing 6.6: Replacing an incorrect value.

```
DELETE { ?o owl:hasValue ?value }
INSERT { ?o owl:hasValue "180.0"^^xsd:decimal }
WHERE {
  ?reel_subclasses rdfs:subClassOf :small_reel.
  ?reel_subclasses ?p ?o.
  ?o owl:hasValue ?value.
  ?o owl:onProperty ?property.
FILTER (?property= :has_outer_diameter && ?value != 180.0)
}
```

Moreover, we leverage SPARQL queries for detecting single construct outliers. First, we identify suitable aggregation levels for the analysis of the target values: packing weight and outer box dimensions (width, length, and height). Therefore, through the distinction and elimination of the outliers within the groups, the acceptable values for each target remain. These values are later used to train the ML model.

Listing 6.7: Detecting the groups with single construct outliers.

```
SELECT ?v1 ?v2 ?v3 ?v4 ?v5 ?v6 ?v7 ?v8
(avg(?target) as ?avg_target)  (min(?target) as ?min_target)
(max(?target) as ?max_target)
WHERE {
  ?v1 ?v2 ?v3 ?v4 ?v5 ?v6 ?v7 ?v8 ?target
  -triple patterns-
}
```

```
GROUP BY ?v1 ?v2 ?v3 ?v4 ?v5 ?v6 ?v7 ?v8
HAVING (
((?avg_target + 3*(?max_target - ?min_target)/4) < ?max_target) ||
((?avg_target - 3*(?max_target - ?min_target)/4) > ?min_target)
)
```

Listing 6.7 illustrates the generic form of the query used for the analysis of each of the target values. A set of variables (?v1 to ?v8) is selected to group the data and the average, minimum and maximum values for the targets are calculated. `-triple pattern-` is the generic form to express *?packing_instances :has_outer_length ?outer_length* or any other triple with properties for the target values. According to our knowledge, there is no formula to calculate the standard deviation directly in a SPARQL query. Therefore, we use an approximation known as "range rule", illustrated in Equation 6.3, where s is the standard deviation approximation. To get the outliers, we apply the common practice of identifying outliers by using three standard deviations away from the mean, Equation 6.4. The HAVING clause includes the conditions to find the outliers, based on a lower and upper thresholds.

$$s = \frac{maximun - minimum}{4} \tag{6.3}$$

$$outliers = average \pm 3 \times s \tag{6.4}$$

The result of this query is the list of those packing instances whose target values exceed or are below the outlier thresholds. We remove the identified outliers by a DELETE clause.

After concluding the data cleaning preparation, we create a data report containing all the concepts in the ontology and the clean data. This data report is the input of the subsequent steps of the proposed system.

Feature Transformation and Selection. Using the report generated in the previous step, the process of transforming and selecting the features begins. Based on the feature type (numerical or categorical), a different analysis takes place. Table 6.6 summarizes the transformations and the measures considered for the realization of this process.

For the feature transformation step, we select the logarithmic and square root transformations done to the numerical features for their effect in reducing the right-skewness and adapting the values to a normal distribution [207, 208]. On the other hand, the categorical features need to be encoded into numerical variables so that the algorithm can understand and learn from them. Therefore, as proposed by Breiman [209] and discussed by Koehrsen and Will [210], these features need a binary encoding, and a commonly used method is One-Hot-Encoding. The logic is to create new independent features for each category of non-numerical features. It then uses a "1" and "0" to denote their presence or absence, respectively.

Step	Numerical features	Categorical features
Transformation	Logarithmic and square root transformations	Binary transformation: One-Hot-Encoding
Selection	Pearson correlation	Density plot and Pearson correlation

Table 6.6: Transformation and selection framework for numerical and categorical features.

For the feature selection step, we choose a backward selection approach guided by the contribution of Guyon and Elisseeff [211]. In an initial step, we calculate the correlations between all the *numerical features* and the targets of the model (packing weight and outer box dimensions). These initial correlation values help distinguish the numerical features that impact the target values, hence affecting the model outcome.

The categorical features, density plots assess the feature relevance before transforming them to numerical values. The density plots help visualize the distributions of the target values by differentiating the categories of the non-numerical features. If the distributions for the different categories of a feature differ, the user can infer that the categorical feature does have an effect on the target value, therefore, it can be considered relevant [212]. For instance, Figure 6.10 is an example of a density plot used to illustrate the distribution of the packing weight. In addition, this distribution is differentiated by the values for the categorical feature reel_size. Based on the different packing weight distributions of the sizes, the user can infer that the size of the packing material reel has an impact on the target packing weight. We create density plots for the remaining categorical features, and based on the distributions observed, only the features considered relevant remained. At the end of this step, We consider only the remaining categorical features for transformation. Then, we transform the chosen features to numerical values and we calculate the correlation between the original and transformed features.

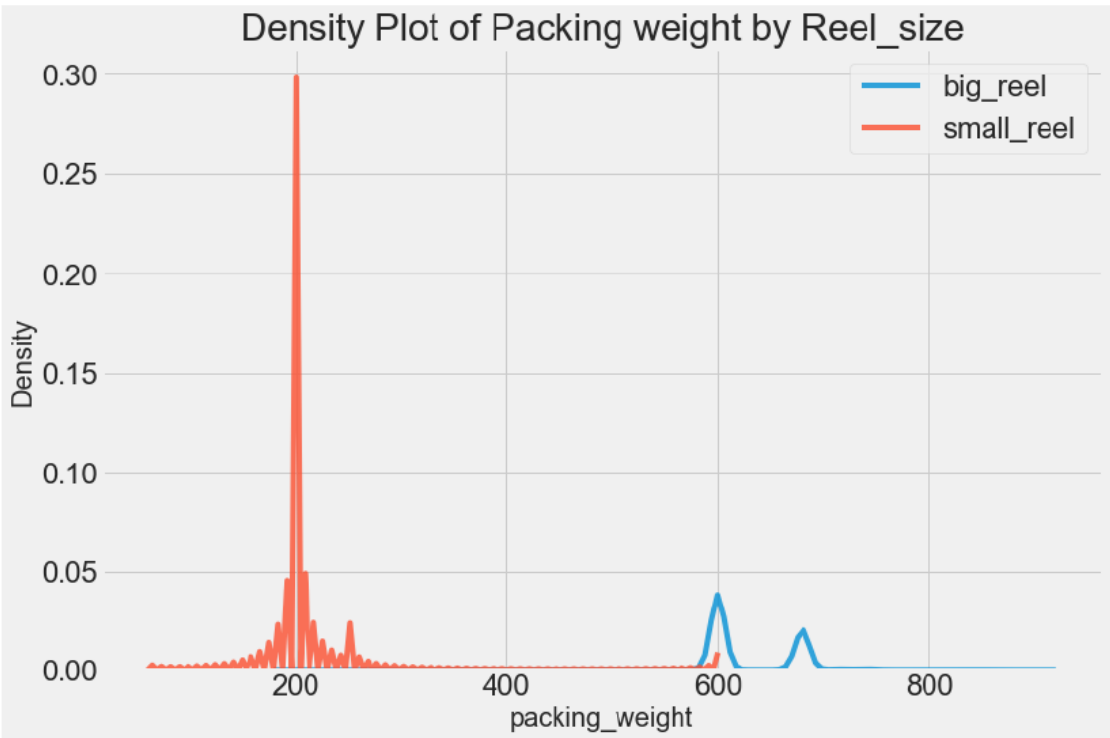


Figure 6.10: Density plot of packing weight showing the effect of the reel_size feature on the packing weight.

Applying this feature transformation and selection method results in an increment in data dimensionality, the initial list of 42 features increased to 1,454 features. The feature selection process, by means of the correlation of Pearson, reduced this quantity to 1,209 features. Then, we remove 245 irrelevant and redundant features. The subsequent step is to generate the training and testing datasets by splitting the complete data. We select the 70-30 split (training-testing) based on the lowest standard error and higher accuracy that this split had in comparison to other evaluated options, as proven by the work of Adelabu et al. [213].

The last step before training the model is **Feature normalization**. We employ the min-max normalization technique as a final step of the data preparation process. We consider a predefined boundary with values between 1 and 0. At the end of this step, we adapt the values of all the features to have a maximum value of 1 and a minimum of 0 based on the study of Shalabi et al. [214] which shows that this method has the best results in terms of accuracy, simplicity and tree-growing time.

Machine Learning Model

The first step towards implementing the ML model is the definition of the type of task required. We identify a regression task based on the use case outcome expectations, i.e., the prediction of numerical values. Moreover, we implement a supervised ML model since the input variables and targets are available.

For selecting the supervised regression model, we compare different algorithms, i.e., Random Forest, Support Vector Machines, K-Nearest Neighbors, Gradient Boosted, and Linear Regression, based on their performance. We select a benchmark of algorithms based on the work of Caruana and Niculescu-Mizil [215] and Yu-Tun et al. [216], as a representative scale of classical supervised learning methods.

We choose Random Forest (RF) because it performs the best without cumbersome parameter optimization, as presented in Table 6.11. This is in line with the state-of-the-art [217]. Additionally, RF models provide a feature relevance output in which the features that are more commonly used on each split are highlighted as we show in section F.2. This output is important since it contributes to the determination of the features used for prediction. Moreover, it is worth mentioning that any ML model is expected to benefit in the same way from clean data, whether the algorithm is further optimized or not. Therefore, the second assumption about the fixed parametrization for this work remains valid.

6.3.2 Evaluation and Discussion

We first evaluate the ontology coverage of the domain incorporated by the competency questions. Then, we show the impact of the ontology-based approach on the prepocessing phase and the ML performance.

Domain Representation

The packing ontology creation has the objective of bringing domain knowledge to the user and showing how an ontology facilitates understanding and feature selection. To evaluate the validity of this hypothesis, we translate the competency questions from Table 6.5 into SPARQL queries and execute to evaluate completeness.

We show in Table 6.7 the query and the result for **CQ4**. Table 6.8 shows an extract of the values retrieved for two different sales products for **CQ5**. As depicted in the tables, it is possible that a

Competency Question

```
SELECT DISTINCT ?packing_material
WHERE {
  :1EBN1001AE rdfs:subClassOf :semiconductor_product;
              rdfs:subClassOf ?a.
  ?a owl:someValuesFrom ?package_name;
     owl:onProperty ?property.
  ?package_name rdfs:subClassOf :package;
                rdfs:subClassOf ?b.
  ?b owl:hasValue ?packing_info;
     owl:onProperty ?property2.
  ?packing_info rdf:type :packing.
  ?c owl:hasValue ?packing_info;
     owl:onProperty ?property3.
  ?material_code rdfs:subClassOf ?c;
                 rdfs:subClassOf :packing_material.
  ?packing_material rdfs:subClassOf :packing_material
     .
}
```

SPARQL Result	packing_material
	moisture_barrier_bag
	reel
	big_reel

Table 6.7: SPARQL query and results of competency question CQ4: "What are the materials used for packing a product?".

sales product is linked to more than one packing method. Therefore, it is possible to have more than one value for packing weight and outer dimensions. There remaining SPARQL queries are in section F.1. Based on the ontology's ability to answer the competency questions, its domain coverage is considered appropriate, domain experts also confirm this. The packing knowledge expressed in the ontology provided the user with the necessary data (of which the packing experts and we are aware of) to implement the ML model.

Data Cleaning Preparation Evaluation

For the subsequent data cleaning steps, the focus is on the detection and elimination of inconsistencies in the data to improve the packing data quality and prepare the data to be used by the ML model. In order to compare the effect of including the SWT, we create a "baseline" preprocessing. For this "baseline" preprocessing, we perform the data understanding and cleaning conventionally through the programming language Python and pandas library for data manipulation, analysis and exploration. Table 6.9 shows an overview of the different types of inconsistencies and the data-specific inconsistencies detected by the ontology-based approach (OB) as well the types of inconsistencies found in the baseline approach (B).

After implementing the data cleaning steps, we find a significant number of inconsistencies. The inconsistency type "Missing values" affect the instances the most. The reasoner can detect

<table>
<tr><td>Competency Question</td><td colspan="3"><pre>
SELECT DISTINCT ?packing_weight
?outer_height ?outer_width ?outer_length
WHERE {
  :1EBN1001AE rdfs:subClassOf :semiconductor_product;
              rdfs:subClassOf ?a.
  ?a owl:someValuesFrom ?package_name;
     owl:onProperty ?property.
  ?package_name rdfs:subClassOf :package;
                rdfs:subClassOf ?b.
  ?b owl:hasValue ?packing_info.
  ?packing_info rdf:type :packing;
                :has_outer_height ?outer_height;
                :has_outer_length ?outer_length;
                :has_outer_width ?outer_width;
                :has_total_packing_weight ?
                    packing_weight.
}
</pre></td></tr>
<tr><td rowspan="3">SPARQL Result</td><td>packing_weight</td><td>outer_height</td><td>outer_width</td></tr>
<tr><td>680</td><td>32</td><td>347</td></tr>
<tr><td>680</td><td>37</td><td>350</td></tr>
</table>

Table 6.8: SPARQL query and results of competency question CQ5: " Which values of packing weight and outer box dimensions does a product have?".

inconsistencies, such as members of disjoint classes. Their correction needs insight into the domain of interest. This knowledge can be acquired during the construction of the ontology. For the "baseline preprocessing approach", we notice that fewer inconsistencies are found. This approach focuses more on the missing values and the single construct outliers. The detection of error outliers proved to be a strength of the ontology, reasoning, and SPARQL queries approach. This strength is based on the packing knowledge that the ontology is able to represent. Based on the significant number of detected and corrected inconsistencies and the shared knowledge regarding those inconsistencies, we conclude that an ontology-based approach is capable of improving the packing data quality.

Feature Selection Evaluation

To measure how the ontology facilitated the feature selection, we compare the features selected from the ontology and query-based preprocessing against the features selected from the “baseline” preprocessing in Table 6.10. For the categorical features, we notice that the two approaches behave similarly. On the contrary, we encounter a main difference in the numerical features. Only the ontological approach includes the feature “devices_per_box”. The importance of this feature is based on its presence in the aggregation level in the analysis of the target values while removing the single construct outliers. Another significant difference is the omission of the “package_body_width” in the “baseline” approach, which, based on the experts’ feedback, is considered important as well. Finally, while the baseline approach considers “devices_per_funcional_packing”, the ontological

Inconsistency	Data-specific inconsistency	OB	B
Missing values	Packing infos without box	X	
Missing values	Empty values for the outer box dimensions	X	X
Missing values	Reel size classification missing	X	
Single construct outliers	Outer box dimensions below/above outlier threshold	X	X
Error outliers	Reel code material classified into in two disjoint classes (big /small size)	X	
Error outliers	Incorrect value for reel outer diameter	X	
Missing values	Empty values for carrier tape width and pitch	X	X
Error outliers	Incorrect packing weight unit	X	X
Single construct outliers	Packing weight values below/above outlier threshold	X	X
Error outliers	Functional packing classified into two disjoint classes (component /dice packing)	X	
Error outliers	Carrier tape width out of acceptable range of values	X	
Missing values	Component sales product missing package name	X	
Single construct outliers	High standard deviations within aggregation levels		X

Table 6.9: Inconsistency detection overview where (OB) is the ontology-based approach and (B) is the baseline approach.

approach includes the logarithmic form of this feature. Up to this point, the features selected after the ontological-based approach are more representative from the experts' point of view.

Machine Learning Algorithm Evaluation

We measure the prediction performance of the ML model by the Mean Absolute Error (MAE) to show how an ontology-based approach in the preprocessing stage affects the ML model. This metric obtains the error from comparing the predicted values against the real values from the test dataset. Table 6.11 shows a comparison of the benchmark models. The RF model has the lowest MAE values for the packing weight, width, and height targets. On the contrary, the length target is better predicted by the Linear Regression (LR). By considering these MAE values as decisive criteria while choosing only one model to predict the four targets, the RF Regression is the model of choice.

Moreover, we compare the prediction error values from the proposed system model against the error values of a "baseline" model. Table 6.12 demonstrates that the MAE for the ontology-based approach is lower for the width, length, and height in comparison to the values of the “baseline” approach. For this last, the packing weight target is better predicted, showing a lower value for MAE.

The ontology-based approach used for the data understanding and cleaning preparation, with the focus on improving the data quality, impacts the selection of features and the model predicting performance. Under this same conclusion, it is considered that the treatment done to the missing values and especially to the error outliers reflects in the reduction of prediction errors for the outer box dimensions (width, length and height). The baseline approach, on the other hand, where a deeper statistical analysis was done of the target values to detect single construct outliers, guided into better

	Feature	OB	B
Categorical features	customer_name	X	X
	package_category	X	X
	package_name	X	X
	package_technology	X	X
	location	X	X
	facility	X	X
	city		X
	functional_packing	X	X
	moisture_protection	X	X
	packing_code_OPN	X	
	box	X	X
	carrier_tape	X	X
	cover_tape	X	X
	moisture_barrier_bag	X	
	reel	X	X
	reel_size	X	X
Numerical Features	package_body_length	X	X
	package_body_width	X	
	package_body_thickness	X	X
	tnid_bom_fit		X
	functional_packing_per_box	X	X
	devices_per_functional_packing		X
	devices_per_box	X	
	max_storage		X
	hub_diameter	X	X
	outer_diameter	X	X
	log_package_body_length	X	X
	log_devices_per_functional_packing	X	

Table 6.10: Feature selection comparison between ontology-based (OB) and baseline (B) preprocessing.

predicting performance for the packing weight target. Python enables deeper statistical analyses which support the detection of groups with high standard deviations. Under this statement, we consider that the features selected in the baseline approach are a better representation of the existing linear relations in regard to the packing weight target.

Discussion

The evaluation of the proposed approach leads to identifying the limitations of SPARQL for deeper statistical methods. It is deemed necessary as a next step to investigate further ways to detect the single construct outliers using semantic technologies.

Regression Model	Packing Weight	Width	Length	Height
Linear	1.68261 e+09	1.09168 e+05	**1.38751 e-02**	8.18766 e+04
Random Forest	**0.75531**	**0.01102**	0.01789	**0.00931**
K-Nearest Neighbors	3.37446	0.19040	0.14427	0.10797
Support Vector Machine	1.93816	0.15209	0.14596	0.12329
Gradient Boosted	8.37769	1.64097	0.44463	0.97918

Table 6.11: Models comparison based on mean absolute error (MAE). The random forest model with the smallest error for the packing weight, width, and height. Linear regression outperforms all other models to predict length.

Approach	Regression Model	Packing Weight	Width	Length	Height
Ontology-based	RF	0.75531	**0.01102**	**0.01789**	**0.00931**
“Baseline”	RF	**0.45657**	0.04379	0.0419	0.0256

Table 6.12: Model predicting performance based on mean absolute error (MAE). Ontology-based approach predicts better the width, length, and height.

6.4 Concluding Remarks

Semiconductor SCs are strongly affected by the bullwhip effect, i.e., increasing demand fluctuation. Understanding the customer’s behavior and needs enable the SC stakeholders to control demand distortion and generate revenue. In this chapter, we proposed KnowGraph-PM, a KG lead-time-based pricing approach allowing tailored revenue generation according to customers’ profiles. With KnowGraph-PM, we enable revenue management strategies to tame the bullwhip effect while reducing the risk of harming relationships with the customers. Moreover, in SCIM-NN, we rely on semantic models, i.e., ontologies and KGs, to structure the domain of context information and COB. Improvements of COB classification help stakeholders like Customer Logistics Managers (CLMs) understand their customers’ behavior and demand better, which is advantageous for long-term production planning. With SCIM-NN, we demonstrate an application of the semantic models to support semiconductor SC challenges via enabling an AI model. Likewise, we implemented an ontology-based approach for data understanding and data preparation phases of a DM process. The created ontology provides the necessary domain understanding; reasoning and a set of SPARQL queries support the preprocessing for an AI algorithm predicting packing information. The proposed system improves the packing data quality of semiconductor products and enables operational excellence.

CHAPTER 7

Conclusion

In this thesis, we study semantic technologies for SCM with a focus on the semiconductor industry. In Chapter 1, we introduced the motivation behind our research. We raised three research questions and provided the plan to address them. Chapter 2 and Chapter 3 gave an overview of the background knowledge and the related work. Chapter 4, Chapter 5 and Chapter 6 highlighted our main contributions and implementations. This chapter summarizes the work and reflects on the concluding answers to the research questions. Also, we show the openings to new research triggered by our work to overcome the identified limitations.

7.1 Research Question Analysis

We revisit the research questions and the proposed contributions.

> RQ1: How can semantic models be used to standardize and benchmark supply chains?

In Chapter 4, we presented SENS, an integrated semantic model of core SC concepts. Also, we provided SENS-GEN, a highly configurable data generator that leverages SENS to create synthetic semantic SC data under various scenario configurations for comprehensive analysis and benchmarking applications. Moreover, we presented the Digital Reference (DR), a standard vocabulary for semiconductor SCs that extends SENS. Afterward, we implemented an approach that relies on ontologies such as DR to ease the building of simulation model for SC analysis.

Thus, we can conclude that semantic models can be used to standardize and benchmark SCs in two ways (1) semantic models for what they are (2) for what they enable.

First, semantic models are well-defined ontologies and KGs that rely on commonly-established SW technology standards, e.g., RDF and OWL. In that sense, creating semantic models for the SC, such as SENS and DR, result in high-level semantics-based descriptions of the domain capturing core artifacts of the E2E SC environment in a standardized way. The output models of our contributions integrate SC concepts, processes, structure, and flows, ensuring an elaborate understanding of the holistic SCs, beyond direct one-to-one relationships and including operational granular details. Hence, semantic models help create operational E2E standardized SCs.

Second, semantic models, being standard SC representations, enable the instantiation of synthetic SCs for simulation and analysis in a systematic way. SENS-GEN and the ontology-based simulation approach, leveraging semantic models (SENS and DR), offer effective means to create empirically controlled and designed SC scenarios. Better simulation and analysis, as put forward by these contributions, help increase the integrated analysis capabilities to standardize and benchmark SCs.

> RQ2: How can semantic data integration help make supply chains more efficient and resilient?

Chapter 5 answers this question and provides two approaches for semantic data integration in SCM to help make SCs more efficient and resilient. First, KnowGraph-MDM is a knowledge-graph-based MDM approach, which relies on establishing a KG layer to represent key MD entities and semantic mappings from and to the original data sources. The output of KnowGraph-MDM is a semantic MDM model that provides a comprehensive definition of SC MD while allowing information integration among SC stakeholders. Consequently, KnowGraph-MDM enables effective SC reporting and decision-making.

Second, semantic data integration implemented in MARE allows the integration of various data sources for disruption management. SC stakeholders can rely on the semantic DMP and resilience evaluation framework in MARE to extract decisions regarding SC structure and operational strategies. MARE facilitates grasping, controlling, and ultimately enhancing SC behavior in complex SC scenarios and increasing the resilience of the supply network.

Hence, semantic data integration, put forward by these contributions, allows information exchange and interoperability to make the SCs more efficient and resilient.

> RQ3: How can we apply semantic technologies to specifically support semiconductor supply chains?

Chapter 6 addresses domain-specific challenges by providing several approaches and applications of semantic technologies for the semiconductor SCs.

First, KnowGraph-PM relies on the fact that manufacturing lead-times of semiconductor products are longer than customer order lead-times. In that sense, KnowGraph-PM leverages a KG and SPARQL to calculate a dynamic price based on integrated order and manufacturing lead-times. KnowGraph-PM allows semiconductor SCs to generate revenue while tailoring to customers' needs and behavior incorporated by respective order lead-times.

Second, semiconductor manufacturers secure a good position in a competitive and volatile market with well-established planning processes. A better understanding of COB enables efficient production planning minimizing inventory excess and critical capacity utilization. Customer context information ensures an enhanced understanding of COB. Thus, we proposed SCIM-NN, which incorporates customer context information, via ontologies and KG embeddings, in a multi-stream neural network classifying COB patterns. Results indicated that representing context information with a KG captures the details of the domain better, thus, improving the overall COB classification.

Similarly, we relied on an ontology-based representation of the domain to enhance the quality of the input data for a ML algorithm predicting the packing information for semiconductor products. The ontology provided the necessary understanding of the domain data, which entailed a positive impact on the data quality, selection of features, and the performance of the model. We can conclude that the ontology-based approach guarantees operational correctness to avoid manufacturing and delivering errors which can be critical in complex SCs such as the semiconductor SC.

To conclude, semantic technologies ensure systematic SCM. Semantic models (e.g, ontologies, KG) provide high-level descriptions of the SC which enable standardization and benchmarking. Semantic data integration allows interoperability and visibility for more efficient and resilient SCs. Semantic technologies can cater to industry-specific characteristics and support domain challenges.

7.2 Future Work

For each chapter, we present the openings to new research, especially to overcome the identified limitations.

Semantic Modeling for Supply Chain Standardization and Benchmarking

We have established with our contributions that semantic models are important for SC standardization, benchmarking, and analysis. Nevertheless, it is important to mention that the evaluation of semantic models is a challenging endeavor. Thus, we list the hurdles to evaluate the proposed semantic models and approaches: SENS, DR, and the methodology to bridge semantic with simulation models.

First, the evaluation of SENS and the presented examples cover the basic flows at this stage. *In Future work*, we propose to further assess SENS and SENS-GEN in light of concrete real-world use cases. The goal will be to validate that SENS can cater to the specific characteristics entailed by the complexity of the manufactured product.

Second, the DR, a semantic vocabulary of the semiconductor SCs, is the output of collaborative ontology development where various domain experts participated. In this contribution, we validate the structural and syntactic correctness of the DR. We also argue that the DR is a semantic reference model for the semiconductor SCs. Future work consists of checking the DR's semantic correctness and completeness. *As next steps*, we propose to evaluate the DR amongst different experts than the ones involved in the DR creation. Otherwise, the SC^3 project goals suggest relying on innovative technologies, e.g., blockchain, to reach consensus among SC stakeholders and validate the proposed model [218].

Moreover, semantic models enable enhancement to the process of building a simulation model as they allow the expansion and the deepening of a domain by the interconnection of models. However, the overall improvement in performance is hardly quantifiable. *Future work* focuses on measuring performance enhancement and the change in efficiency after introducing the SW. This can be measured by the time to construct a simulation model using legacy techniques instead of using the rule-based engine. This is challenging as the time taken to reach a simulation model can only be roughly estimated by the simulation engineer to come to the model in terms of design and creating it using a suitable tool. Furthermore, we propose to test the scalability of the ontology-based simulation methodology by using the DR to automate the creation of E2E SC simulation models.

Semantic Data Integration for Supply Chain Applications

In this thesis, we propose various SC applications that leverage semantic data integration to increase efficiency and resilience. The proposed applications focus on MDM and DMP. *As an outlook*, we can examine the portability of the created methodologies in further SC pillars, e.g., customer relationship management. For instance, we can test the portability of KnowGraph-MDM to reach a full semantic data integration for MDM including customer and industry-specific behavior. Likewise, we propose to use MARE in order to incorporate further SC artifacts, e.g., decision-makers and responsible stakeholders and examine the impact of their involvement on recovery strategies.

Further, including the decision-making chain, process owners, and SC organization structures in our methodologies implies more practicality and closes the gap between theory and practice. Hence, we suggest studying the role of SC stakeholders in deploying the proposed applications in the SCM architecture and enterprise systems.

Semantic technologies for Semiconductor Industry

We proposed semantic-based applications that address semiconductor industry specifics. We deem that semantic models are easily expandable. Hence, applications relying on such models can continuously extend to relevant domains and potentially enhance their performance. For instance, adding contextual data about the customer, e.g., Customer Master Data (address data, contract data) will allow further pricing models to incorporate more factors about a customer. Similarly, more context information like holidays, disruptions, or trade sanctions should be integrated into the context model while predicting customer order behavior. Also, product-related context like life cycle information as well as extensive market-related context information indicating economic up- or downturns can be added.

7.3 Implications

We demonstrate the impact of our work on SCs strategies. Then, we provide technical reports to prove the reproducibility of our work. We discuss the implications of our work on the use of semantic technologies as an enabler for AI models.

Supply Chain Strategic Implications Semantic models, e.g., SENS and DR, resemble digital twins, that facilitate information exchange and integration, hence, allowing an optimized control in complex SC scenarios [219]. The structural and operational information integration in the overall SC enabled by our work increases visibility. This, in turn, may lead to dramatically reducing demand distortion, i.e., the bullwhip effect [220] and strategic positioning an organization in the supply network. Semantic-based SC applications, e.g., MARE are used to simulate the SC behavior under various complex scenarios. SC stakeholders can make informed decisions based on the performance analysis to redesign into a more resilient SC coping with unexpected events. Moreover, proposed applications, i.e., KnowGraph-PM and SCIM-NN, allow SC customer-driven approaches by tailoring operational strategies (revenue management and production planning) to fit customers' profiles, needs, and contexts.

Re-usability of Semantic Supply Chain Models Semantic modeling provides a human and machine-understandable representation of the domain. Therefore, we see implications of the mentioned ontology-based models on the re-usability of SC models. Other SC modeling research areas, e.g., Supply Chain Formation (SCF) and simulation can rely on SENS and DR to ease the extraction of SC configurations for SCF [221] or to standardize the creation of simulation models as proposed by [148].

Reproducibility of our Work We provide the detailed technical report of the code for section 4.1 and section 5.2 in Appendix B and Appendix D, respectively. This documentation shows the reproducibility of our work as the results can be achieved again and the proposed outlook can be implemented.

Semantic Technologies as Enabler for Artificial Intelligence Models SCIM-NN and ontology-based preprocessing show the impact of leveraging ontologies and KGs to feed an ML model. Consequently, we elevate semantic models to a mature data structure surrounded by a useful stack of technologies and tools (reasoning, SPARQL, triple stores, and visualization). All of the mentioned features present the SWT as enablers for AI models for different SC analysis.

Bibliography

[1] V. Soundararajan, S. Sahasranamam, Z. Khan, and T. Jain, “Multinational enterprises and the governance of sustainability practices in emerging market supply chains: An agile governance perspective,” *Journal of World Business*, vol. 56, no. 2, p. 101149, 2021.

[2] D. M. Lambert and M. C. Cooper, “Issues in supply chain management,” *Industrial marketing management*, vol. 29, no. 1, pp. 65–83, 2000.

[3] A. Velayutham, A. R. Rahman, A. Narayan, and M. Wang, “Pandemic turned into pandemonium: the effect on supply chains and the role of accounting information,” *Accounting, Auditing & Accountability Journal*, vol. 34, no. 4, 2021.

[4] M. G. Attinasi, R. De Stefani, E. Frohm, V. Gunnella, G. Koester, M. Tóth, A. Melemenidis *et al.*, “The semiconductor shortage and its implication for euro area trade, production and prices,” *Economic Bulletin Boxes*, vol. 4, no. 1, 2021.

[5] A. Winkelmann, S. Fleischer, S. Herwig, and J. Becker, “A conceptual modeling approach for supply chain event management (scem),” *ECIS 2009 Proceedings*, vol. 2, 2009.

[6] T. M. Simatupang and R. Sridharan, “A benchmarking scheme for supply chain collaboration,” *Benchmarking: An International Journal*, vol. 11, no. 1, pp. 9–31, 2004.

[7] Infineon Technologies, “Infineon annual report,” 2021, unpublished technical report.

[8] N. Ramzy, “Sens: Semantic e2e supply chain generator,” Apr. 2021. [Online]. Available: https://doi.org/10.5281/zenodo.6451155

[9] N. Ramzy and E. Beitz, “Mare: Semantic supply chain disruption management,” Apr. 2021. [Online]. Available: https://doi.org/10.5281/zenodo.6451242

[10] SCC, “Supply chain operations reference (scor) model: Overview – version 10.0,” Supply Chain Council, Cypress, TX, Report, 2010, accessed on 01.09.2021. [Online]. Available: https://www.bibsonomy.org/bibtex/2ff5ce5e6ee7ee359ca001c98bd1d8cf9/flint63

[11] D. Irfan, X. Xiaofei, and D. S. Chun, “A scor reference model of the supply chain management system in an enterprise.” *International Arab Journal of Information Technology (IAJIT)*, vol. 5, no. 3, 2008.

[12] M. S. Golan, L. H. Jernegan, and I. Linkov, “Trends and applications of resilience analytics in supply chain modeling: systematic literature review in the context of the covid-19 pandemic,” *Environment Systems and Decisions*, vol. 40, pp. 222–243, 2020.

[13] T. Bier and A. Lange, “A formation model for supply networks: a fundament for investigations of complex supply networks,” *European Conference on Information Systems 2018*, vol. 1, no. 139, 2018.

[14] J. Rice and R. Hoppe, “Supply chain versus supply chain: The hype & the reality,” *Supply Chain Management Review*, vol. 5, no. 5, pp. 46–54, 2001.

[15] G. E. Delipinar and B. Kocaoglu, “Using scor model to gain competitive advantage: A literature review,” *Procedia-Social and Behavioral Sciences*, vol. 229, pp. 398–406, 2016.

[16] A. Brintrup and A. Ledwoch, “Supply network science: Emergence of a new perspective on a classical field,” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 28, no. 3, p. 033120, 2018.

[17] R. Xiao, Z. Cai, and X. Zhang, “An optimization approach to cycle quality network chain based on improved scor model,” *Progress in Natural Science*, vol. 19, no. 7, pp. 881–890, 2009.

[18] S. H. Huan, S. K. Sheoran, and G. Wang, “A review and analysis of supply chain operations reference (scor) model,” *Supply chain management: An international Journal*, vol. 9, no. 1, p. 23–29, 2004.

[19] C.-F. Chien, S. Dauzère-Pérès, H. Ehm, J. W. Fowler, Z. Jiang, S. Krishnaswamy, T.-E. Lee, L. Mönch, and R. Uzsoy, “Modelling and analysis of semiconductor manufacturing in a shrinking world: challenges and successes,” *European Journal of Industrial Engineering 4*, vol. 5, no. 3, pp. 254–271, 2011.

[20] J. P. Kleijnen, “Supply chain simulation tools and techniques: a survey,” *International journal of simulation and process modelling*, vol. 1, no. 1-2, pp. 82–89, 2005.

[21] H. Min, “Artificial intelligence in supply chain management: theory and applications,” *International Journal of Logistics: Research and Applications*, vol. 13, no. 1, pp. 13–39, 2010.

[22] R. Toorajipour, V. Sohrabpour, A. Nazarpour, P. Oghazi, and M. Fischl, “Artificial intelligence in supply chain management: A systematic literature review,” *Journal of Business Research*, vol. 122, pp. 502–517, 2021.

[23] R. Dash, M. McMurtrey, C. Rebman, and U. K. Kar, “Application of artificial intelligence in automation of supply chain management,” *Journal of Strategic Innovation and Sustainability*, vol. 14, no. 3, pp. 43–53, 2019.

[24] I. M. Cavalcante, E. M. Frazzon, F. A. Forcellini, and D. Ivanov, “A supervised machine learning approach to data-driven simulation of resilient supplier selection in digital manufacturing,” *International Journal of Information Management*, vol. 49, pp. 86–97, 2019.

[25] K. N. Amirkolaii, A. Baboli, M. Shahzad, and R. Tonadre, “Demand forecasting for irregular demands in business aircraft spare parts supply chains by using artificial intelligence (ai),” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 15 221–15 226, 2017.

[26] S. Nawab, S. Teggihalli *et al.*, “Performance optimization for extraction, transformation, loading and reporting of data,” in *2015 Global Conference on Communication Technologies (GCCT)*. IEEE, 2015, pp. 516–519.

[27] “Ready for anything? turbulence and the resilience imperative,” February 2021, [published 26.02.2021, accessed 01.10.2021]. [Online]. Available: https://www.eiu.com/n/ready-for-anything-turbulence-and-the-resilience-imperative/

[28] H. A. Smith and J. D. McKeen, “Developments in practice xxx: master data management: salvation or snake oil?” *Communications of the Association for Information Systems*, vol. 23, no. 1, p. 4, 2008.

[29] F. Lindmark, “Master data management: Creating a common language for master data across an extended and complex supply chain,” Masterarbeit, Linköping University, Department of Computer and Information Science, Software and Systems, 2018.

[30] M. Millar, *Global supply chain ecosystems: strategies for competitive advantage in a complex, connected world*. Kogan Page Publishers, 2015.

[31] B. Otto, “Functional reference architecture for corporate master data management,” Ph.D. dissertation, University of St. Gallen, 2009.

[32] J. Kokemüller and A. Weisbecker, “Master data management: Products and research.” in *Institut Català d’Investigació Química (ICIQ)*. Citeseer, 2009, pp. 8–18.

[33] Infineon Technologies, “Introduction of semantic web and ontologies for md analytics (sewo4mda) - development and evaluation of a prototype,” 2021.

[34] J.-M. Israël, “Item ii. 1 the integrated reporting framework,” 2018, accessed on: 04.05.2020. [Online]. Available: https://www.integratedreporting.org/wp-content/uploads/2021/01/InternationalIntegratedReportingFramework.pdf

[35] P. Russom, “Seven reasons why master data management needs data governance,” *TDWI Checklist Report. TDWI Research*, August 2012.

[36] R. Vilminko-Heikkinen, P. Brous, and S. Pekkola, “Paradoxes, conflicts and tensions in establishing master data management function,” in *24th European Conference on Information Systems, ECIS 2016*, 2016.

[37] D. Loshin, *Master data management*. Morgan Kaufmann, 2010.

[38] A. Dreibelbis, E. Hechler, I. Milman, M. Oberhofer, P. van Run, and D. Wolfson, *Enterprise Master Data Management (Paperback): An SOA Approach to Managing Core Information*. Pearson Education, 2008.

[39] Z. Murti, A. Andarrachmi, A. N. Hidayanto, and S. B. Yudhoatmojo, “Master data management planning:(case study of personnel information system at xyz institute),” in *2018 International Conference on Information Management and Technology (ICIMTech)*. IEEE, 2018, pp. 160–165.

[40] A. I. Alharbi, “Master data management,” *Journal of Information Systems Technology and Planning*, vol. 8, no. 19, pp. 95 - 101, 2016.

[41] B. Otto and A. Schmidt, “Enterprise master data architecture: Design decisions and options.” in *International Conference on Information Quality (ICIQ)*, 2010, p. 147–159.

[42] D. Fitzpatrick, F. Coallier, and S. Ratté, “A holistic approach for the architecture and design of an ontology-based data integration capability in product master data management,” in *IFIP International Conference on Product Lifecycle Management*. Springer, 2012, pp. 559–568.

[43] R. Winter and R. Fischer, “Essential layers, artifacts, and dependencies of enterprise architecture,” in *2006 10th IEEE International Enterprise Distributed Object Computing Conference Workshops (EDOCW’06)*. IEEE, 2006, pp. 30–30.

[44] B. Otto and A. Reichert, “Organizing master data management: findings from an expert survey,” in *Proceedings of the 25th ACM Symposium on Applied Computing, Sierre, Switzerland*, 2010.

[45] C. W. Craighead, J. Blackhurst, M. J. Rungtusanatham, and R. B. Handfield, “The severity of supply chain disruptions: design characteristics and mitigation capabilities,” *Decision sciences*, vol. 38, no. 1, pp. 131–156, 2007.

[46] J. Blackhurst, C. W. Craighead, D. Elkins, and R. B. Handfield, “An empirically derived agenda of critical research issues for managing supply-chain disruptions,” *International journal of production research*, vol. 43, no. 19, pp. 4067–4081, 2005.

[47] J. R. Macdonald and T. M. Corsi, “Supply chain disruption management: Severe events, recovery, and performance,” *Journal of Business Logistics*, vol. 34, no. 4, pp. 270–288, 2013.

[48] N. Bugert and R. Lasch, “Supply chain disruption models: A critical review,” *Logistics Research*, vol. 11, no. 5, pp. 1–35, 2018.

[49] A. Ismail and H. Ehm, “Simulating and evaluating supply chain disruptions along and end-end semiconductor automotive supply chain amid covid-19 crisis,” in *AnyLogic Conference Workshop Proceedings*, 2021.

[50] F.-M. Jaenichen, C. J. Liepold, A. Ismail, C. J. Martens, V. Dörrsam, and H. Ehm, “Simulating and evaluating supply chain disruptions along an end-end semiconductor automotive supply chain,” in *Proceedings of the winter simulation conference*, vol. 1. IEEE, 2021.

[51] D. Simchi-Levi, W. Schmidt, Y. Wei, P. Y. Zhang, K. Combs, Y. Ge, O. Gusikhin, M. Sanders, and D. Zhang, “Identifying risks and mitigating disruptions in the automotive supply chain,” *Interfaces*, vol. 45, no. 5, pp. 375–390, 2015.

[52] F. Sabouhi, M. S. Pishvaee, and M. S. Jabalameli, “Resilient supply chain design under operational and disruption risks considering quantity discount: A case study of pharmaceutical supply chain,” *Computers & Industrial Engineering*, vol. 126, pp. 657–672, 2018.

[53] D. Ernst, “The economics of electronics industry: competitive dynamics and industrial organization,” *The International Encyclopedia of Business and Management (IEBM)*, 2000.

[54] B. Aytac and S. D. Wu, “Characterization of demand for short life-cycle technology products,” *Annals of Operations Research*, vol. 203, no. 1, pp. 255–277, 2013.

[55] H. Ehm and T. Ponsignon, “Future research directions for mastering end-to-end semiconductor supply chains,” in *2012 IEEE International Conference on Automation Science and Engineering (CASE)*. IEEE, 2012, pp. 641–645.

[56] H. T. Grahn, *Introduction to semiconductor physics*. World Scientific Publishing Company, 1999.

[57] J. Aelker, T. Bauernhansl, and H. Ehm, “Managing complexity in supply chains: A discussion of current approaches on the example of the semiconductor industry,” *Procedia CIRP*, vol. 7, pp. 79–84, 2013.

[58] B. A. Mousavi, R. Azzouz, C. Heavey, and H. Ehm, “Simulation-based analysis of the nervousness within semiconductors supply chain planning: insight from a case study,” in *2019 Winter Simulation Conference (WSC)*. IEEE, 2019, pp. 2396–2407.

[59] A. Seitz, H. Ehm, R. Akkerman, and S. Osman, “A robust supply chain planning framework for revenue management in the semiconductor industry,” *Journal of Revenue and Pricing Management*, vol. 15, no. 6, pp. 523–533, 2016.

[60] E. Sucky, “The bullwhip effect in supply chains—an overestimated problem?” *International Journal of Production Economics*, vol. 118, no. 1, pp. 311–322, 2009.

[61] J. Dai, S. Li, and S. Peng, “Analysis on causes and countermeasures of bullwhip effect,” in *MATEC web of conferences*, vol. 100. EDP Sciences, 2017, p. 05018.

[62] T. Ponsignon, “The bullwhip effect in the semiconductor industry on the example of infineon technologies,” Masterarbeit, Université franco-allemande, 2006.

[63] T. de Kok, F. Janssen, J. Van Doremalen, E. Van Wachem, M. Clerkx, and W. Peeters, “Philips electronics synchronizes its supply chain to end the bullwhip effect,” *Interfaces*, vol. 35, no. 1, pp. 37–48, 2005.

[64] G. Stefansson, “Business-to-business data sharing: A source for integration of supply chains,” *International journal of production economics*, vol. 75, no. 1-2, pp. 135–146, 2002.

[65] F. Chen, Z. Drezner, J. K. Ryan, and D. Simchi-Levi, “Quantifying the bullwhip effect in a simple supply chain: The impact of forecasting, lead times, and information,” *Management science*, vol. 46, no. 3, pp. 436–443, 2000.

[66] M. D. Wittman and P. P. Belobaba, “Customized dynamic pricing of airline fare products,” *Journal of Revenue and Pricing Management*, vol. 17, no. 2, pp. 78–90, 2018.

[67] I. Deksnyte and Z. Lydeka, “Dynamic pricing and its forming factors,” *International Journal of Business and Social Science*, vol. 3, no. 23, 2012.

[68] C. Gibbs, D. Guttentag, U. Gretzel, L. Yao, and J. Morton, “Use of dynamic pricing strategies by airbnb hosts,” *International Journal of Contemporary Hospitality Management*, vol. 30, no. 1, pp. 2–20, 2018.

[69] H. Chen, O. Q. Wu, and D. D. Yao, "On the benefit of inventory-based dynamic pricing strategies," *Production and Operations Management*, vol. 19, no. 3, pp. 249–260, 2010.

[70] H. Ott, S. Heilmayer, C. Sng *et al.*, "Granularity dependency of forecast accuracy in semiconductor industry," *Research in Logistics & Production*, vol. 3, 2013.

[71] M. Öner-Közen and H. Ehm, "Dynamic price and lead time quotation under semiconductor industry related challenges," in *2018 Winter Simulation Conference (WSC)*. IEEE, 2018, pp. 3386–3396.

[72] N. Berger and A. Strauss, "Order prioritization in the high-tech industry: A lead-time based pricing approach," 2021, unpublished research paper: accessed 2021.

[73] J. Wirtz, S. E. Kimes, J. H. P. Theng, and P. Patterson, "Revenue management: resolving potential customer conflicts," *Journal of Revenue and Pricing Management*, vol. 2, no. 3, pp. 216–226, 2003.

[74] V. Kumar, "Customer relationship management," *Wiley international encyclopedia of marketing*, 2010.

[75] D. S. Johnson, B. H. Clark, and G. Barczak, "Customer relationship management processes: How faithful are business-to-business firms to customer profitability?" *Industrial Marketing Management*, vol. 41, no. 7, pp. 1094–1105, 2012.

[76] R. Thakur and L. Workman, "Customer portfolio management (cpm) for improved customer relationship management (crm): Are your customers platinum, gold, silver, or bronze?" *Journal of Business Research*, vol. 69, no. 10, pp. 4095–4102, 2016.

[77] M. Zhang and P. Bell, "Price fencing in the practice of revenue management: An overview and taxonomy," *Journal of Revenue and Pricing Management*, vol. 11, no. 2, pp. 146–159, 2012.

[78] H. Yli-Renko and R. Janakiraman, "How customer portfolio affects new product development in technology-based entrepreneurial firms," *Journal of marketing*, vol. 72, no. 5, pp. 131–148, 2008.

[79] M. Ratusny, M. Schiffer, and H. Ehm, "Customer order behavior classification via convolutional neural networks in the semiconductor industry," 2022, unpublished manuscript.

[80] N. Garcia, B. Renoust, and Y. Nakashima, "Context-aware embeddings for automatic art analysis," in *Proceedings of the 2019 on International Conference on Multimedia Retrieval*, A. El Saddik, A. Del Bimbo, Z. Zhang, A. Hauptmann, K. S. Candan, M. Bertini, L. Xie, and X.-Y. Wei, Eds. New York, NY, USA: ACM, 2019, pp. 25–33.

[81] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," *Advances in neural information processing systems*, vol. 27, 2014.

[82] Z. Tu, W. Xie, J. Dauwels, B. Li, and J. Yuan, "Semantic cues enhanced multimodality multistream cnn for action recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 5, pp. 1423–1437, 2019.

[83] H. Wang and L. Wang, “Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 499–508.

[84] S. Jain, C.-C. Lim, B.-P. Gan, and Y.-H. Low, “Criticality of detailed modeling in semiconductor supply chain simulation,” in *Proceedings of the 31st conference on Winter simulation: Simulation—a bridge to the future-Volume 1*, 1999, pp. 888–896.

[85] L. Mönch, R. Uzsoy, and J. W. Fowler, “A survey of semiconductor supply chain models part i: semiconductor supply chains, strategic network design, and supply chain simulation,” *International Journal of Production Research*, vol. 56, no. 13, pp. 4524–4545, 2018.

[86] C. Sun and T. Rose, “Supply chain complexity in the semiconductor industry: assessment from system view and the impact of changes,” *IFAC-PapersOnLine*, vol. 48, no. 3, pp. 1210–1215, 2015.

[87] J. W. Fowler, L. Mönch, and T. Ponsignon, “Discrete-event simulation for semiconductor wafer fabrication facilities: A tutorial.” *International Journal of Industrial Engineering*, vol. 22, no. 5, 2015.

[88] Y. H. Lee, S. Chung, B. Lee, and K. H. Kang, “Supply chain model for the semiconductor industry in consideration of manufacturing characteristics,” *Production Planning & Control*, vol. 17, no. 5, pp. 518–533, 2006.

[89] J. Lin, V. L. Spiegler, and M. M. Naim, “Dynamic analysis and design of a semiconductor supply chain: a control engineering approach,” *International Journal of Production Research*, vol. 56, no. 13, pp. 4585–4611, 2018.

[90] H. Ehm, H. Wenke, L. Mönch, T. Ponsignon, and L. Forstner, “Towards a supply chain simulation reference model for the semiconductor industry,” in *Proceedings of the 2011 Winter Simulation Conference (WSC)*. IEEE, 2011, pp. 2119–2130.

[91] P. Hitzler, M. Krötzsch, S. Rudolph, and Y. Sure, *Semantic Web: Grundlagen*. Springer-Verlag, 2007.

[92] M. Cheatham and C. Pesquita, “Semantic data integration,” in *Handbook of big data technologies*. Springer, 2017, pp. 263–305.

[93] M. Gagnon, “Ontology-based integration of data sources,” in *2007 10th International Conference on Information Fusion*. IEEE, 2007, pp. 1–8.

[94] G. De Giacomo, D. Lembo, M. Lenzerini, A. Poggi, and R. Rosati, “Using ontologies for semantic data integration,” in *A Comprehensive Guide Through the Italian Database Research Over the Last 25 Years*. Springer, 2018, pp. 187–202.

[95] S. P. Gardner, “Ontologies and semantic data integration,” *Drug discovery today*, vol. 10, no. 14, pp. 1001–1007, 2005.

[96] M. Caridi, A. Moretto, A. Perego, and A. Tumino, "The benefits of supply chain visibility: A value assessment model," *International Journal of Production Economics*, vol. 151, pp. 1–19, 2014.

[97] H.-L. Wei and E. T. Wang, "The strategic value of supply chain visibility: increasing the ability to reconfigure," *European Journal of Information Systems*, vol. 19, no. 2, pp. 238–249, 2010.

[98] K. Pal, "Integrating heterogeneous enterprise data using ontology in supply chain management," in *Big Data and Knowledge Sharing in Virtual Organizations*. IGI Global, 2019, pp. 71–102.

[99] Y. Ye, D. Yang, Z. Jiang, and L. Tong, "Ontology-based semantic models for supply chain management," *The International Journal of Advanced Manufacturing Technology*, vol. 37, no. 11-12, pp. 1250–1260, 2008.

[100] T. Grubic and I.-S. Fan, "Supply chain ontology: Review, analysis and synthesis," *Computers in Industry*, vol. 61, no. 8, pp. 776–786, 2010.

[101] B. Jachimczyk, R. Tkaczyk, T. Piotrowski, S. Johansson, and W. Kulesza, "Iot-based dairy supply chain-an ontological approach," *Elektronika ir Elektrotechnika*, vol. 27, no. 1, pp. 71–83, 2021.

[102] Q. Long, K. Song, and S. Yang, "Semantic modeling for the knowledge framework of computational experiments and decision making for supply chain networks," *IEEE Access*, vol. 7, pp. 46 363–46 375, 2019.

[103] A. G. Suherman and T. M. Simatupang, "The network business model of cloud computing for end-to-end supply chain visibility," *International Journal of Value Chain Management*, vol. 8, no. 1, pp. 22–39, 2017.

[104] M. Zdravković, H. Panetto, M. Trajanović, and A. Aubry, "An approach for formalising the supply chain operations," *Enterprise Information Systems*, vol. 5, no. 4, pp. 401–421, 2011.

[105] 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," in *2016 IEEE Tenth International Conference on Semantic Computing (ICSC)*. IEEE, 2016, pp. 132–139.

[106] 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*, vol. 26, no. 1-2, pp. 140–151, 2013.

[107] M. Fayez, L. Rabelo, and M. Mollaghasemi, "Ontologies for supply chain simulation modeling," in *Proceedings of the Winter Simulation Conference, 2005.* IEEE, 2005, pp. 7–20.

[108] F. Persson, "Scor template—a simulation based dynamic supply chain analysis tool," *International Journal of Production Economics*, vol. 131, no. 1, pp. 288–294, 2011.

[109] M. Li, F. Yang, R. Uzsoy, and J. Xu, "A metamodel-based monte carlo simulation approach for responsive production planning of manufacturing systems," *Journal of Manufacturing Systems*, vol. 38, pp. 114–133, 2016.

[110] J. Yuan and T. Ponsignon, “Towards a semiconductor supply chain simulation library (scsc-simlib),” in *Proceedings of the Winter Simulation Conference 2014*. IEEE, 2014, pp. 2522–2532.

[111] H. Ewen, L. Moench, H. Ehm, T. Ponsignon, J. W. Fowler, and L. Forstner, “A testbed for simulating semiconductor supply chains,” *IEEE Transactions on Semiconductor Manufacturing*, vol. 30, no. 3, pp. 293–305, 2017.

[112] G. Laipple, S. Dauzère-Pérès, T. Ponsignon, and P. Vialletelle, “Generic data model for semiconductor manufacturing supply chains,” in *2018 Winter Simulation Conference (WSC)*. IEEE, 2018, pp. 3615–3626.

[113] M. Hassoun, D. Kopp, L. Mönch, and A. Kalir, “A new high-volume/low-mix simulation testbed for semiconductor manufacturing,” in *2019 Winter Simulation Conference (WSC)*. IEEE, 2019, pp. 2419–2428.

[114] P. Benjamin, M. Patki, and R. Mayer, “Using ontologies for simulation modeling,” in *Proceedings of the 2006 Winter Simulation Conference*. IEEE, 2006, pp. 1151–1159.

[115] Y. M. Teo and C. Szabo, “Codes: An integrated approach to composable modeling and simulation,” in *41st Annual Simulation Symposium (anss-41 2008)*. IEEE, 2008, pp. 103–110.

[116] G. A. Silver, L. W. Lacy, and J. A. Miller, “Ontology based representations of simulation models following the process interaction world view,” in *Proceedings of the 2006 winter simulation conference*. IEEE, 2006, pp. 1168–1176.

[117] A. Soares, A. Azevedo, J. P. De Sousa *et al.*, “Distributed planning and control systems for the virtual enterprise: organizational requirements and development life-cycle,” *Journal of Intelligent Manufacturing*, vol. 11, no. 3, pp. 253–270, 2000.

[118] R. Herding and L. Mönch, “Designing an ontology for agent-based planning and control tasks in semiconductor supply chains,” in *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. Springer, 2016, pp. 65–75.

[119] S. Auer, H. Kim, S. Scerri, and M. Galkin, Eds., *Integration strategies for enterprise knowledge graphs*. Tenth IEEE International Conference on Semantic Computing, ICSC 2016, Laguna Hills, USA, 2016. [Online]. Available: 10.1109/ICSC.2016.24

[120] J. H. Lee, S. J. Fenves, C. Bock, H.-W. Suh, S. Rachuri, X. Fiorentini, and R. D. Sriram, “A semantic product modeling framework and its application to behavior evaluation,” *IEEE Transactions on Automation Science and Engineering*, vol. 9, no. 1, pp. 110–123, 2011.

[121] C. Yu, F.-p. Zhang, S. I. Butt, Y. Yan, and W. Lv, “Ontoimm: An ontology for product intelligent master model,” *Applied Sciences*, vol. 9, no. 12, p. 2553, 2019.

[122] X. Wang, X. Sun, F. Cao, L. Ma, N. Kanellos, K. Zhang, Y. Pan, and Y. Yu, “Smdm: enhancing enterprise-wide master data management using semantic web technologies,” *Proceedings of the VLDB Endowment*, vol. 2, no. 2, pp. 1594–1597, 2009.

[123] D. Ivanov and A. Dolgui, “A digital supply chain twin for managing the disruption risks and resilience in the era of industry 4.0,” *Production Planning & Control*, pp. 1–14, 2020.

[124] S. Emmenegger, E. Laurenzini, and B. Thönssen, “Improving supply-chain-management based on semantically enriched risk descriptions.” in *KMIS*, 2012, pp. 70–80.

[125] C. Palmer, E. N. Urwin, A. Niknejad, D. Petrovic, K. Popplewell, and R. I. Young, “An ontology supported risk assessment approach for the intelligent configuration of supply networks,” *Journal of Intelligent Manufacturing*, vol. 29, no. 5, pp. 1005–1030, 2018.

[126] C. S. Singh, G. Soni, and G. K. Badhotiya, “Performance indicators for supply chain resilience: review and conceptual framework,” *Journal of Industrial Engineering International*, vol. 15, no. 1, pp. 105–117, 2019.

[127] R. Wirth and J. Hipp, “Crisp-dm: Towards a standard process model for data mining,” in *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*, vol. 1. Springer-Verlag London, UK, 2000, pp. 29–39.

[128] D. Dou, H. Wang, and H. Liu, “Semantic data mining: A survey of ontology-based approaches,” in *Proceedings of the 2015 IEEE 9th international conference on semantic computing (IEEE ICSC 2015).* IEEE, 2015, pp. 244–251.

[129] T. R. Gruber, “A translation approach to portable ontology specifications,” *Knowledge acquisition*, vol. 5, no. 2, pp. 199–220, 1993.

[130] P. Ristoski and H. Paulheim, “Semantic web in data mining and knowledge discovery: A comprehensive survey,” *Journal of Web Semantics*, vol. 36, pp. 1–22, 2016.

[131] I. E. Gonzalez, “Semantic technologies for supporting kdd processes,” Ph.D. dissertation, PhD thesis, Universidad del País Vasco-Euskal Herriko Unibertsitatea, 2019.

[132] E. Rahm and H. H. Do, “Data cleaning: Problems and current approaches,” *IEEE Data Eng. Bull.*, vol. 23, no. 4, pp. 3–13, 2000.

[133] S. Grimm, A. Abecker, J. Völker, and R. Studer, “Ontologies and the semantic web,” in *Domingue J., Fensel D., Hendler J.A. (eds) Handbook of Semantic Web Technologies*. Springer, Berlin, Heidelberg, 2011, pp. 507–579.

[134] D. Kontokostas, P. Westphal, S. Auer, S. Hellmann, J. Lehmann, R. Cornelissen, and A. Zaveri, “Test-driven evaluation of linked data quality,” in *Proceedings of the 23rd international conference on World Wide Web*, 2014, pp. 747–758.

[135] C. Fürber and M. Hepp, “Towards a vocabulary for data quality management in semantic web architectures,” in *Proceedings of the 1st International Workshop on Linked Web Data Management*, 2011, pp. 1–8.

[136] C. Fürber and M. Hepp, “Using semantic web technologies for data quality management,” in *Handbook of Data Quality*. Springer, Berlin, Heidelberg, 2013, pp. 141–161.

[137] S. Brüggemann and F. Gruening, "Using domain knowledge provided by ontologies for improving data quality management," in *Proceedings of I-Know*. Citeseer, 2008, pp. 251–258.

[138] X. Wang, H. J. Hamilton, and Y. Bither, *An ontology-based approach to data cleaning*. Department of Computer Science, University of Regina Regina, Canada, 2005.

[139] D. Perez-Rey, A. Anguita, and J. Crespo, "Ontodataclean: Ontology-based integration and preprocessing of distributed data," in *Biological and Medical Data Analysis. ISBMDA*. Springer, Berlin, Heidelberg, 2006, pp. 262–272.

[140] Y. Dai, S. Wang, N. N. Xiong, and W. Guo, "A survey on knowledge graph embedding: Approaches, applications and benchmarks: Electronics, 9(5), 750," *Electronics*, vol. 9, no. 5, p. 750, 2020.

[141] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," *Advances in Neural Information Processing Systems*, vol. 26, 2013.

[142] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," in *International conference on machine learning*. PMLR, 2016, pp. 2071–2080.

[143] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. d. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier *et al.*, "Knowledge graphs," *Synthesis Lectures on Data, Semantics, and Knowledge*, vol. 12, no. 2, pp. 1–257, 2021.

[144] A. García-Durán and M. Niepert, "KBLRN : End-to-end learning of knowledge base representations with latent, relational, and numerical features," *CoRR*, vol. abs/1709.04676, 2017. [Online]. Available: http://arxiv.org/abs/1709.04676

[145] A. Kristiadi, M. A. Khan, D. Lukovnikov, J. Lehmann, and A. Fischer, "Incorporating literals into knowledge graph embeddings," in *International Semantic Web Conference*. Springer, 2019, pp. 347–363.

[146] N. Ramzy, S. Auer, J. Chamanara, and H. Ehm, "Sens: Semantic synthetic benchmarking model for integrated supply chain simulation and analysis," *European Conference on Information Systems 2022*, vol. 1, 2022.

[147] H. Ehm, N. Ramzy, P. Moder, C. Summerer, S. Fetz, and C. Neau, "Digital reference–a semantic web for semiconductor manufacturing and supply chains containing semiconductors," in *2019 Winter Simulation Conference (WSC)*. IEEE, 2019, pp. 2409–2418.

[148] N. Ramzy, C. J. Martens, S. Singh, T. Ponsignon, and H. Ehm, "First steps towards bridging simulation and ontology to ease the model creation on the example of semiconductor industry," in *2020 Winter Simulation Conference (WSC)*. IEEE, 2020, pp. 1789–1800.

[149] E. Schragenheim, H. W. Dettmer, and J. W. Patterson, *Supply chain management at warp speed: Integrating the system from end to end*. CRC Press, 2009.

[150] A. Brintrup, Y. Wang, and A. Tiwari, “Supply networks as complex systems: a network-science-based characterization,” *IEEE Systems Journal*, vol. 11, no. 4, pp. 2170–2181, 2015.

[151] D. Arndt and J. Broekstra, “Rdf-star and sparql-star,” W3C, W3C Community, Dec. 2021, accessed on 01.11.2021. [Online]. Available: https://w3c.github.io/rdf-star/cg-spec/editors_draft

[152] P. Patel-Schneider and P. Hayes, “RDF 1.1 semantics,” W3C, W3C Recommendation, Feb. 2014, accessed on 01.09.2021. [Online]. Available: https://www.w3.org/TR/2014/REC-rdf11-mt-20140225/

[153] J. S. Hofstetter and J. H. Grimm, “Multi-tier sustainable supply chain management,” in *Handbook on the Sustainable Supply Chain*. Edward Elgar Publishing Cheltenham/Northampton, 2019, p. 526–540.

[154] T. Xiao and X. Qi, “A two-stage supply chain with demand sensitive to price, delivery time, and reliability of delivery,” *Annals of Operations Research*, vol. 241, no. 1, pp. 475–496, 2016.

[155] B. Borgström and S. Hertz, “Supply chain strategies: Changes in customer order-based production,” *Journal of Business Logistics*, vol. 32, no. 4, pp. 361–373, 2011.

[156] S. Cannella, R. Dominguez, B. Ponte, and J. M. Framinan, “Capacity restrictions and supply chain performance: Modelling and analysing load-dependent lead times,” *International Journal of Production Economics*, vol. 204, pp. 264–277, 2018.

[157] M. Setak, S. Sharifi, and A. Alimohammadian, “Supplier selection and order allocation models in supply chain management: a review,” *World applied sciences journal*, vol. 18, no. 1, pp. 55–72, 2012.

[158] 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,” in *2016 IEEE Tenth International Conference on Semantic Computing (ICSC)*. IEEE, 2016, pp. 132–139.

[159] G. Thoma, J. Popp, D. Nutter, D. Shonnard, R. Ulrich, M. Matlock, D. S. Kim, Z. Neiderman, N. Kemper, C. East *et al.*, “Greenhouse gas emissions from milk production and consumption in the united states: A cradle-to-grave life cycle assessment circa 2008,” *International Dairy Journal*, vol. 31, pp. S3–S14, 2013.

[160] W. W. W. C. (W3C), “The organization ontology,” 2014, accessed 19.04.2021. [Online]. Available: https://www.w3.org/TR/vocab-org/

[161] W3C, “Time ontology in owl,” 2020, accessed on 19.04.2021. [Online]. Available: https://www.w3.org/TR/owl-time/

[162] N. F. Noy and M. A. Musen, “The prompt suite: interactive tools for ontology merging and mapping,” *International journal of human-computer studies*, vol. 59, no. 6, pp. 983–1024, 2003.

[163] N. Ramzy, H. Ehm, S. Dusrst, K. Wibmer, and W. Bick, “Knowgraph-tt: Knowledge-graph-based transit time matching in semiconductor supply chains,” *Infocommunications Journal: A Publication of the Scientific Association for Infocommunications (HTE)*, vol. 14, no. 1, 2022. [Online]. Available: https://doi.org/10.36244/ICJ.2022.1.7

[164] B. Glimm, I. Horrocks, B. Motik, G. Stoilos, and Z. Wang, “Hermit: an owl 2 reasoner,” *Journal of Automated Reasoning*, vol. 53, no. 3, pp. 245–269, 2014.

[165] Knowledge Representation and Reasoning Group, “Hermit owl reasoner,” http://www.hermit-reasoner.com/, 2022, accessed on 22.02.2022.

[166] M. A. Musen, “The protégé project: a look back and a look forward,” *AI Matters*, vol. 1, no. 4, pp. 4–12, 2015. [Online]. Available: https://doi.org/10.1145/2757001.2757003

[167] J. Morecroft, S. Robinson *et al.*, “Explaining puzzling dynamics: comparing the use of system dynamics and discrete-event simulation,” in *Proceedings of the 23rd International Conference of the System Dynamics Society*, vol. 115. System Dynamics Society Boston, MA, 2005.

[168] A. A. Tako and S. Robinson, “Model development in discrete-event simulation and system dynamics: An empirical study of expert modellers,” *European Journal of Operational Research*, vol. 207, no. 2, pp. 784–794, 2010.

[169] G. A. Silver, J. A. Miller, M. Hybinette, G. Baramidze, and W. S. York, “An ontology for discrete-event modeling and simulation,” *Simulation*, vol. 87, no. 9, pp. 747–773, 2011.

[170] H. Kaiya and M. Saeki, “Using domain ontology as domain knowledge for requirements elicitation,” in *14th IEEE International Requirements Engineering Conference (RE’06)*. IEEE, 2006, pp. 189–198.

[171] X. H. Wang, D. Q. Zhang, T. Gu, and H. K. Pung, “Ontology based context modeling and reasoning using owl,” in *IEEE annual conference on pervasive computing and communications workshops, 2004. Proceedings of the second*. IEEE, 2004, pp. 18–22.

[172] S. I. GmbH, “Cluster configuration for plasma etching & deposition.” SENTECH Instruments GmbH, Tech. Rep., 2016, accessed on 30.07.2020. [Online]. Available: https://www.sentech.com/en/Cluster-Configuration__300/

[173] N. Ramzy, S. Durst, and M. Schreiber, “Knowgraph-mdm: A methodology for knowledge-graph-based master data management,” *24th IEEE International Conference on Business Informatics*, 2022, in workshop proceedings.

[174] N. Ramzy, S. Auer, H. Ehm, and J. Chamanara, “Mare: Semantic supply chain disruption management and resilience evaluation framework,” in *24th International Conference on Enterprise Information Systems (ICEIS)*. Springer, 2022.

[175] N. F. Noy, D. L. McGuinness *et al.*, “Ontology development 101: A guide to creating your first ontology,” 2001, accessed on 15.12.2020. [Online]. Available: http://protege.stanford.edu/publications/ontologydevelopment/ontology101.pdf

[176] M. Smajevic and D. Bork, “From conceptual models to knowledge graphs: a generic model transformation platform,” in *2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*. IEEE, 2021, pp. 610–614.

[177] F. Ekaputra, M. Sabou, E. Serral Asensio, E. Kiesling, and S. Biffl, “Ontology-based data integration in multi-disciplinary engineering environments: A review,” *Open Journal of Information Systems*, vol. 4, no. 1, pp. 1–26, 2017.

[178] A. A. Detoni, G. M. Miranda, L. D. Renault, J. P. A. Almeida, R. de Almeida Falbo, G. Guizzardi, F. A. Baião, and R. S. Guizzardi, “Where enterprise architecture and early ontology engineering meet: A case study in the public security domain.” in *JOWO*, 2017.

[179] D. Wiśniewski, J. Potoniec, A. Ławrynowicz, and C. M. Keet, “Analysis of ontology competency questions and their formalizations in sparql-owl,” *Journal of Web Semantics*, vol. 59, p. 100534, 2019.

[180] C. Bezerra, F. Freitas, and F. Santana, “Evaluating ontologies with competency questions,” in *2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, vol. 3. IEEE, 2013, pp. 284–285.

[181] Stanford University, “Protege - a free, open-source ontology editor and framework for building intelligent systems,” https://protege.stanford.edu/, 2021, accessed 27.04.2021. [Online]. Available: https://protege.stanford.edu/

[182] Eccenca GmbH, “The entreprise knowledge graph platform,” 2021, accessed on 03.01.2022. [Online]. Available: https://eccenca.com/products/enterprise-knowledge-graph-platform-corporate-memory

[183] J. Hardi, M. O’Connor, and T. Tudorache, “Cellfie-plugin,” 2021, accessed on 28.02.2022. [Online]. Available: https://github.com/protegeproject/cellfie-plugin

[184] J. Lehmann and L. Bühmann, “Autosparql: Let users query your knowledge base,” in *Extended semantic web conference*. Springer, 2011, pp. 63–79.

[185] P. Ochieng, “Parot: Translating natural language to sparql,” *Expert Systems with Applications*, vol. 176, p. 114712, 2021.

[186] D. Simbizi, L. Benabbou, and B. Urli, “Systematic literature reviews in supply chain resilience: A systematic literature review,” in *11th Annual International Conference on Industrial Engineering and Operations Management, IEOM 2021*, 2021, pp. 327–340.

[187] T. Sawik, “A multi-portfolio approach to integrated risk-averse planning in supply chains under disruption risks,” in *Handbook of Ripple Effects in the Supply Chain*. Springer, 2019, pp. 35–63.

[188] D. Messina, A. C. Barros, A. L. Soares, and A. Matopoulos, “An information management approach for supply chain disruption recovery,” *The International Journal of Logistics Management*, no. 31, p. 489–519, 2020.

[189] O. Burkacky, S. Lingemann, and K. Pototzky, “Coping with the auto-semiconductor shortage: Strategies for success,” McKinsey & Company, Tech. Rep., 2021, accessed on 05.10.2021. [Online]. Available: https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/coping-with-the-auto-semiconductor-shortage-strategies-for-success

[190] B. Tomlin and Y. Wang, “Operational strategies for managing supply chain disruption risk,” *The handbook of integrated risk management in global supply chains*, pp. 79–101, 2011.

[191] S. K. Paul, S. Asian, M. Goh, and S. A. Torabi, “Managing sudden transportation disruptions in supply chains under delivery delay and quantity loss,” *Annals of Operations Research*, vol. 273, no. 1-2, pp. 783–814, 2019.

[192] T. Sawik, “Two-period vs. multi-period model for supply chain disruption management,” *International Journal of Production Research*, vol. 57, no. 14, pp. 4502–4518, 2019.

[193] C. A. MacKenzie, K. Barker, and J. R. Santos, “Modeling a severe supply chain disruption and post-disaster decision making with application to the japanese earthquake and tsunami,” *IIE Transactions*, vol. 46, no. 12, pp. 1243–1260, 2014.

[194] H. Carvalho, V. Cruz-Machado, and J. G. Tavares, “A mapping framework for assessing supply chain resilience,” *International Journal of Logistics Systems and Management*, vol. 12, no. 3, pp. 354–373, 2012.

[195] N. Ramzy, S. Auer, J. Chamanara, and H. Ehm, “Knowgraph-pm: A knowledge graph based pricing model for semiconductor supply chains,” in *International Conference on Intelligence Science*. Springer, 2021, pp. 61–75.

[196] P. Ulirch, N. Ramzy, and M. Ratusny, “Scim-nn: Semantic context informationmodeling for neural networks in customer order behavior classification.” in *EEE Transactions on Semiconductor Manufacturing 2022, Special Issue on Production-LevelArtificial Intelligence Applications in Semiconductor Manufacturing*. IEEE.

[197] P. C. Soto, N. Ramzy, F. Ocker, and B. Vogel-Heuser, “An ontology-based approach for preprocessing in machine learning,” in *2021 IEEE 25th International Conference on Intelligent Engineering Systems (INES)*. IEEE, 2021, pp. 133–138.

[198] X. L. Wang and R. Brennan, “A framework for key account management and revenue management integration,” *Industrial Marketing Management*, vol. 43, no. 7, pp. 1172–1181, 2014.

[199] C. Li, “Controlling the bullwhip effect in a supply chain system with constrained information flows,” *Applied Mathematical Modelling*, vol. 37, no. 4, pp. 1897–1909, 2013.

[200] N. Berger, “Revenue management in the semiconductor industry,” Masterarbeit, WHU Otto Beisheim School of Management, 2017.

[201] W3C, “Defining n-ary relations on the semantic web,” https://www.w3.org/TR/swbp-n-aryRelations/, 2006, accessed on 10.01.2022.

[202] Wikidata, "Main page," https://www.wikidata.org/wiki/Wikidata:Main_Page, 2019, accessed on 10.01.2022.

[203] G. A. Gesese, R. Biswas, M. Alam, and H. Sack, "A survey on knowledge graph embeddings with literals: Which model links better literal-ly?" *Semantic Web*, vol. 12, no. 4, pp. 617–647, 2021.

[204] J.-B. Lamy, "Owlready: Ontology-oriented programming in python with automatic classification and high level constructs for biomedical ontologies," *Artificial intelligence in medicine*, vol. 80, pp. 11–28, 2017.

[205] M. Ali, M. Berrendorf, C. T. Hoyt, L. Vermue, S. Sharifzadeh, V. Tresp, and J. Lehmann, "Pykeen 1.0: a python library for training and evaluating knowledge graph embeddings," *Journal of Machine Learning Research*, vol. 22, no. 82, pp. 1–6, 2021.

[206] H. Aguinis, R. K. Gottfredson, and H. Joo, "Best-practice recommendations for defining, identifying, and handling outliers," *Organizational Research Methods*, vol. 16, no. 2, pp. 270–301, 2013.

[207] D. Ruppert, "Transformation of data," in *The International Encyclopedia of the Social and Behavioral Sciences*. Elsevier Amsterdam, 2001.

[208] J. Osborne, "Improving your data transformations: Applying the box-cox transformation," *Practical Assessment, Research, and Evaluation*, vol. 15, no. 1, p. 12, 2010.

[209] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[210] Koehrsen and Will, "A complete machine learning project walk-through in python: Part one," in *A Complete Machine Learning Project Walk-Through in Python: Part One*. Apress, May 2018, accessed on 15.11.2020. [Online]. Available: https://towardsdatascience.com/a-complete-machine-learning-walk-through-in-python-part-one-c62152f39420

[211] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of machine learning research*, vol. 3, no. Mar, pp. 1157–1182, 2003.

[212] O. C. Wilke, "Fundamentals of data visualization: a primer on making informative and compelling figures," in *Fundamentals of data visualization: a primer on making informative and compelling figures*. O'Reilly Media, Inc, 2019, pp. 65–69.

[213] S. Adelabu, O. Mutanga, and E. Adam, "Testing the reliability and stability of the internal accuracy assessment of random forest for classifying tree defoliation levels using different validation methods," *Geocarto International*, vol. 30, no. 7, pp. 810–821, 2015.

[214] L. Al Shalabi, Z. Shaaban, and B. Kasasbeh, "Data mining: A preprocessing engine," *Journal of Computer Science*, vol. 2, no. 9, pp. 735–739, 2006.

[215] R. Caruana and A. Niculescu-Mizil, "An empirical comparison of supervised learning algorithms," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 161–168.

[216] A. Y. T. Wang, R. J. Murdock, S. K. Kauwe, A. O. Oliynyk, A. Gurlo, J. Brgoch, K. A. Persson, and T. D. Spark, “Machine learning for materials scientists: An introductory guide toward best practices,” in *Chemistry of Materials*. ACS Publications, 2020, vol. 32, pp. 4954–4965.

[217] T. Hastie, R. Tibshirani, and J. H. Friedman, “Data mining, inference, and prediction,” in *The Elements of Statistical Learning*, 2nd ed. New York: Springer, 2009.

[218] H. Ehm, E. Schoitsch, J. W. van der Weit, N. Ramzy, L. Luo, and D. L. Gruetzner, “Digital reference: a quasi-standard for digitalization in the domain of semiconductor supply chains,” in *2020 IEEE Conference on Industrial Cyberphysical Systems (ICPS)*, vol. 1. IEEE, 2020, pp. 563–570.

[219] S. Y. Barykin, A. A. Bochkarev, E. Dobronravin, and S. M. Sergeev, “The place and role of digital twin in supply chain management,” *Academy of Strategic Management Journal*, vol. 20, pp. 1–19, 2021.

[220] Y. Blomkvist and L. Ullemar Loenbom, “Improving supply chain visibility within logistics by implementing a digital twin: A case study at scania logistics,” KTH Institute of Technology: Stockholm, Sweden, 2020.

[221] F. Ameri and B. Kulvatunyou, “Modeling a supply chain reference ontology based on a top-level ontology,” in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 59179. American Society of Mechanical Engineers, 2019, p. V001T02A052.

[222] R. Quante, H. Meyr, and M. Fleischmann, “Revenue management and demand fulfillment: matching applications, models and software,” in *Supply Chain Planning*. Springer, 2009, pp. 57–88.

APPENDIX A

List of Publications

The following is a complete chronological list of the contributions completed during the work on this thesis:

Conference Papers:

1. Patrick Moder, Hans Ehm, **Nour Ramzy**, *Digital Twin for Plan and Make Using Semantic Web Technologies – Extending the JESSI/SEMATECH MIMAC Standard to the Digital Reference*. In Proceedings of Digital Transformation in Semiconductor Manufacturing, pp. 24-32. Springer, 2019;

2. Hans Ehm, **Nour Ramzy**, Patrick Moder, Christoph Summerer, Simone Fetz, and Cédric Neau. *Digital Reference–A Semantic Web for Semiconductor Manufacturing and Supply Chains Containing Semiconductors*. In 2019 Winter Simulation Conference (WSC), pp. 2409-2418. IEEE, 2019;

3. **Nour Ramzy**, Christian James Martens, Shreya Singh, Thomas Ponsignon, and Hans Ehm.*First Steps Towards Bridging Simulation And Ontology To Ease The Model Creation On The Example Of Semiconductor Industry*. In 2020 Winter Simulation Conference (WSC), pp. 1789-1800. IEEE, 2020;

4. Hartwig Baumgärtel, Patrick Moder, **Nour Ramzy**, Hans Ehm. *Service-based Semiconductor Manufacturing using the Digital Reference Ontology for Global Service Discovery*. In IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society, pp. 4533-4540. IEEE, 2020;

5. Hans Ehm, Erwin Schoitsch, Jan Wytze van der Weit, **Nour Ramzy**, Lanyingzhu Luo, Daniel Louis Gruetzner. *Digital Reference: a quasi-standard for digitalization in the domain of semiconductor supply chains*, In 2020 IEEE Conference on Industrial Cyberphysical Systems (ICPS), vol. 1, pp. 563-570. IEEE, 2020;

6. **Nour Ramzy**, Sören Auer, Javad Chamanara, Hans Ehm. *KnowGraph-PM: A Knowledge-Graph-based Pricing Model for Semiconductor Supply Chains*. In 8th International Conference on on Intelligence Science, pp. 61-75. Springer, 2021;

7. Patricia Centeno Soto, **Nour Ramzy**, Felix Ocker, Birgit Vogel-Heuser. *An ontology-based approach for preprocessing in machine learning: use case for packing material information*. In 25th IEEE International Conference on Intelligent Engineering Systems, pp. 133-138. IEEE, 2021.

8. **Nour Ramzy**, Sören Auer, Javad Chamanara, and Hans Ehm. *Sens: Semantic synthetic benchmarking model for integrated supply chain simulation and analysis*. In 30th European Conference on Information Systems, 2022;

9. **Nour Ramzy**, Sören Auer, Javad Chamanara, Hans Ehm. *MARE: Semantic Supply Chain Disruption Management and Resilience Evaluation Framework*. In 24th International Conference on Enterprise Information Systems (ICEIS), 2022.

10. **Nour Ramzy**, Philipp Ulrich, Lancelot Mairesse, Hans Ehm. *Demand Predictability Evaluation for Supply Chain Processes Using Semantic Web Technologies*. Under review In 2019 Winter Simulation Conference (WSC), 2022.

Journal Papers:

11. **Nour Ramzy** , Hans Ehm, S. Dusrst, K. Wibmer, and W. Bick. *Knowgraph-tt:Knowledge-graph-based transit time matching in semiconductor supply chains*. In Infocommunications Journal: A Publication of the Scientific Association forInfocommunications (HTE), vol. 14, pp. 51-58, 2022;

12. Philipp Ulrich, **Nour Ramzy**, Marco Ratusny. *SCIM-NN: Semantic Context Information modeling for Neural Networks in Customer Order Behavior Classification*. Under review in IEEE Transactions on Semiconductor Manufacturing, Special Issue on Production-Level Artificial Intelligence Applications in Semiconductor Manufacturing, 2022;

Workshops and Special Sessions:

13. **Nour Ramzy,** Hans Ehm, Vitalis Wiens, Laura Kohnen. *The Digital Reference: Semantically Connecting Semiconductor Supply Chains to Customers-The Open Online Sales and Marketing Vision*. In 2021 IEEE 17th International Conference on Automation Science and Engineering, Special session Collaborative ontologies for semiconductor Supply chains, IEEE, 2021.

14. Yun Ti, Patrick Moder, **Nour Ramzy**, Hans Ehm. *Investigating Semantic Web as Enabler for Semiconductor Supply Chain Collaboration*. In 2021 IEEE 17th International Conference on Automation Science and Engineering, Special session Collaborative ontologies for semiconductor Supply chains, IEEE, 2021.

15. **Nour Ramzy**, Sandra Durst Martin Schreiber, Sören Auer *KnowGraph-MDM: A Methodology for Knowledge-Graph-based Master Data Management* . Under review in The workshop proceedings for24th IEEE International Conference on Business Informatics, IEEE, 2022.

Book Chapter:

16. Hans Ehm, Hartwig Baumgärtel , Patrick Moder, Fabian Steinemann, Christoph Summerer, **Nour Ramzy**. *Semantic Web: Befähiger der Industrie 4.0* In Handbuch Industrie 4.0, 2019.

APPENDIX B

SENS and SENS-GEN: Technical Report

This section is a detailed technical report where we describe the installation, reproducibility and coding details of SENS and SENS-GEN. First, we introduce the installation process and the source code accessibility. Also, we present the input and output resources involved and the code structure. Then, we explain the reproducibility of SENS and SENS-GEN in terms of usability and extendability.

B.1 Installation

The code is accessible by a pull command of the project from the repository `https://github.com/NourRamzy/E2E_SC` into an integrated development environment (IDE). The SENS-GEN code is set as a maven project that contains all the libraries and dependencies required to run the JAVA project. The project is linked to a readme file with all the source code. The project is available under the DOI *10.5281/zenodo.5675085*.

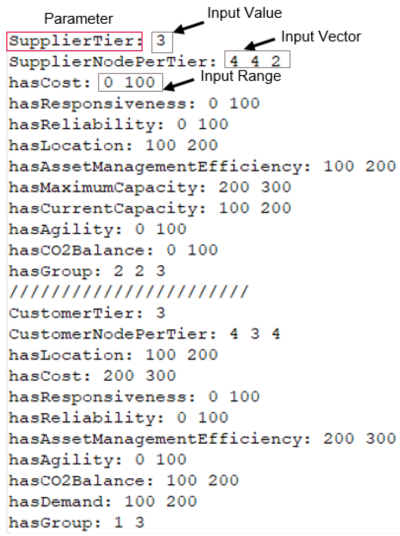
B.2 Resources

In the source code, we create a resources folder, i.e., *src/main/resources* where we define the input and the outputs resources for SENS-GEN.

B.2.1 Input

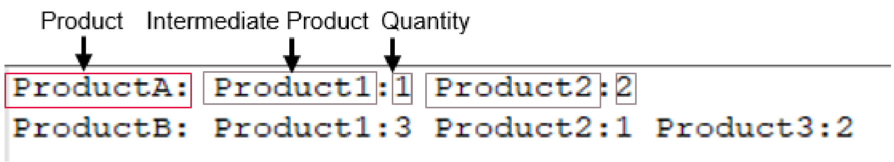
SENS-GEN relies on three input files to run and behave as described:

- **SENS Ontology:** is the semantic model defined in section 4.1.1, used by SENS-GEN to create an synthetic instance of a SC. The current version of the ontology is stored in *src/main/resources/generator.owl* as an OWL file. We assign `http://www.semanticweb.org/ramzy/ontologies/2021/3/untitled-ontology-6` as a local prefix for the ontology as it is not publicly published yet.
- **Parametrization Input:** contains the parameters and their corresponding values for SENS-GEN as described in Table 4.1. In the source code, this file is located in *src/main/resources/configurationfile.txt*. We show a sample of how the file looks like in Figure B.1.




```
SupplierTier: 3
SupplierNodePerTier: 4 4 2
hasCost: 0 100
hasResponsiveness: 0 100
hasReliability: 0 100
hasLocation: 100 200
hasAssetManagementEfficiency: 100 200
hasMaximumCapacity: 200 300
hasCurrentCapacity: 100 200
hasAgility: 0 100
hasCO2Balance: 0 100
hasGroup: 2 2 3
//////////////////////
CustomerTier: 3
CustomerNodePerTier: 4 3 4
hasLocation: 100 200
hasCost: 200 300
hasResponsiveness: 0 100
hasReliability: 0 100
hasAssetManagementEfficiency: 200 300
hasAgility: 0 100
hasCO2Balance: 100 200
hasDemand: 100 200
hasGroup: 1 3
```

Figure B.1: Input file detailing SENS-GEN parameterization with parameters and corresponding values to determine the topology of the generated synthetic supply chain.




```
ProductA: Product1:1 Product2:2
ProductB: Product1:3 Product2:1 Product3:2
```

Figure B.2: List of products, intermediate products and corresponding quantities.

- **Products Input:** defines the products manufactured by the SC as modeled by the triple *Node manufactures Product*. This file located in *src/main/resources/products.txt* details the intermediate products and the corresponding quantities required to manufacture a product as modeled by the triple *«P needsProduct ?comp» needsQuantity ?quant*. Figure B.2 shows an example of how this input file is structured.

B.2.2 Output

- After running the code of SENS-GEN, the output is SENS KG stored in *src/main/java/output.ttl*. In Figure 4.2, we show an example of the output KG of an automotive SC generated via the input parameters.

- SENS-GEN is capable of evaluating the performance of the instantiated KG. The output values of the benchmarking process show on the IDE console and indicate the performance of the SC in the experimental setup.

B.3 Code Structure

1. Algorithm 1 in section 4.1.2 shows the steps to generate a KG relying on the input parameters and the SENS ontology. This algorithm consists of the following methods:
   a) `create_OEM()`: this function creates one instance of the class OEM and sets the values for the following properties: *hasDeliveryTime, hasTransportMode, hasInventory* and the corresponding characteristics of an inventory *hasProduct, hasCost, hasQuantity, hasTimeStamp*.
   b) `create_tiers_nodes()`: this function consists of `create_Supplier()` as well as `create_Customer()` methods to generate the SC nodes and corresponding tiers based on the input parameters.
   c) `create_relations()`: after the execution of this method nodes are connected via *hasUpStreamNode, hasDownStreamNode* while tiers are linked with *hasUpStreamTier, hasDownStreamTier*.
   d) `generation()`: this function generates the initial values for capacity, inventory, saturation for all nodes. Also, via `create_orders()` we assign orders to customer nodes and corresponding products, delivery times and quantities, i.e., *hasProduct, hasDeliveryTime, hasQuantity*.

2. `fulfillDemand()`: implements the logic for demand fulfillment described in section 4.1.1. After the execution of this function supply, there is a supply plan specific for each order modeled by *Order hasSupplyPlan SupplyPlan*.

3. `evaluationMetrics()`: implements the benchmarking and integrated analysis in experimental contexts, enabled by SENS-GEN section 4.1.3. We provide in *Evaluation_KPI/* folder various SPARQL-based performance indicators, e.g., utilization

B.4 Reproducibility

With the current status of SENS-GEN code, we can instantiate a synthetic SC. By changing the input parameters, one can tailor the structure where the topology corresponds to an industry sector as it signifies the complexity of the products (the steps needed to manufacture), the variability and the number of customers and suppliers. The behavior of the SC nodes changes by modifying the input properties, e.g., *hasReliability, hasCO2Balance*. Using the existing KPIs as SPARQL queries in section 4.1.3, we can evaluate the performance of the synthetic SC designed.

By reproducing the actual state of SENS and SENS-GEN, we can also extend and modify the code to:

- change SENS model in *src/main/resources/generator.owl*. We can edit SENS ontology locally by adding data properties to the nodes to model additional node performance characteristics such as carbon footprint, service level and price. This will enable the implementation of a multi-factor-based supplier choice.
- modify the product catalogue of the SC detailed in *src/main/resources/products.txt*.

- implement a different demand fulfillment model. We propose a backward scheduling model for demand fulfillment, while [222] proposes other approaches to fulfill customers' demand.
- add evaluation metrics. We propose a sample of KPIs to measure the SC performance. Including a SPARQL-based metric in *Evaluation_KPI/* allows to study and benchmark the performance relying on desired metrics.

in

APPENDIX C

KnowGraph-MDM: A Methodology for Knowledge-Graph-based Master Data Management

In this Section, we show the translated competency questions into SPARQL as part of the semantic conceptualization step in KnowGraph-MDM. We present sample results not included in the contribution.

Competency Question

```
PREFIX smi: <http://exampleURI/KG_MDM#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT  DISTINCT ?var ?report
WHERE {
  ?relation rdfs:domain ?var.
  ?relation rdfs:range ?report.

FILTER(regex(str(?var),'Sales_Nr') || regex(str(?var)
    ,'PR_Nr') ).
}
```

SPARQL Result

Var	Report
Sales_Nr	ReportA
Sales_Nr	ReportB
PR_Nr	ReportC
Sales_Nr	ReportC
Sales_Nr	ReportD

Table C.1: SPARQL query and results of competency question 2.2: "In which data report is a specific set of Data Fields contained? – OR Relationship".

Competency Question

```
PREFIX smi: <http://http://exampleURI/KG_MDM #>
PREFIX owl:<http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT  DISTINCT ?relation ?comment ? ?minCardinality ?maxCardinality
WHERE {
  ?relation rdfs:domain smi:Sales_Nr.
  ?relation rdfs:comment ?comment.
  ?relation rdfs:range ?restriction.

  ?restriction owl:onClass smi:Production_Nr.
  ?restriction owl:onProperty smi:assignedTo_ProductionNr.
  ?restriction owl:minQualifiedCardinality ?minQualifiedCardinality.

  ?restriction2 owl:onClass smi:Production_Nr.
  ?restriction2 owl:onProperty smi:assignedTo_ProductionNR.
  ?restriction2 owl:maxQualifiedCardinality ?maxQualifiedCardinality.
}
```

SPARQL Result	Relation	Comment	min Cardinality	max Cardinality
	assignedTo _ProductionNR	Relationship between Sales_Nr and Production_NR is a 1:N relationship.	0	1000000

Table C.2: SPARQL Query and Results of Competency Question 4:"What is the cardinality restriction describing the relationship between different Data Fields?" .

Competency Question

```
PREFIX smi: <http://http://exampleURI/KG_MDM#>
SELECT DISTINCT ?div ?bu ?pl ?hfg ?fgr
WHERE {
  ?sales_nr smi:assigned_SPName ?sp_name.
  FILTER(regex(str(?sp_name),'10804DA')).
  ?sp_nr smi:has_RFP ?rfp.
  ?rfp smi:has_FGR ?fgr.
  ?fgr smi:has_HFG ?hfg.
  ?hfg smi:has_PL ?pl.
  ?pl smi:has_BL ?bu.
  ?bl smi:has_DIV ?div.
}
```

SPARQL Result	DIV	BU	PL	HFG	FGR
	PSS	HIR	19	Z	Z04

Table C.3: SPARQL query and results of competency question 5: "Which are the Business Segments and Product Groups of a specific Product? ".

APPENDIX **D**

MARE: Technical Report

In this section, we provide a detailed technical report where we describe the installation, reproducibility and coding details for MARE. First, we explain the code accessibility and installation steps. Then, we present the input and output files for MARE. Also, we describe the structure of the code that allows the reproducibility of MARE.

D.1 Installation

The code for MARE is available on the github repository `https://github.com/NourRamzy/MARE--Resilience-Framework` as a maven project, also under the DOI *10.5281/zenodo.6451242*. A simple pull request into a JAVA-supporting IDE makes MARE accessible and editable.

D.2 Resources

MARE relies on various inputs to implement semantic disruption management. MARE generates as an output an evaluation for the resilience aggregated by SC or by customers. We store the input and output of MARE in *src/main/resources*.

D.2.1 Input

In order to examine the disruptions effect, we rely on the data generated and provided by the synthetic generator described in the technical report [8]. We assume the pre-existence of a synthetic SC with the orders, their corresponding supply plans and time frames. The input file is in *src/resources/supplychain.ttl*. We provide the disruption ontology in *src/resources/disruption.owl* to detail the characteristics of a disruption and create instantiated disruption events.

D.2.2 Output

MARE incorporates an evaluation framework for SC resilience and recovery. We rely on the recovery performance evaluation metrics described in section 5.2.3 to compare the pre-disruption supply plans to the recovered supply generated in the recovery phase. The result of the evaluation shows on the IDE console and indicates the performance of the SC in the experimental setup. Moreover, the code generates *src/resources/supplychain_output.ttl*, an output KG file that incorporates the input synthetic SC from *src/resources/supplychain.ttl* as well as all generated artifacts of MARE, e.g., the disruption KG, the disrupted and recovered supply plans as per Figure 5.3.

D.3 Code Structure

The structure of the code reflects the steps of MARE to model, assess , recover and evaluate disruptions and SC resilience as detailed in section 5.2.

- **Model**: in order to model disruptions we create the method `create_disruptions()`. It implements the required disruptions with the corresponding characteristcs, e.g., *hasScope, hasSeverity, hasBeginDate, hasEndDate* and *hasLocation*. This function relies on the disruption ontology model in *src/resources/disruption.owl*.
- **Assess**: As described in MARE, in order to assess the consequences of disruptions, first, we retrieve the affected nodes, i.e., fall within the disruption location and time frame. The method `get_disrupted_plans()` consists of the SPARQL query to retrieve the disrupted supply plans for the affected nodes as shown in Listing 5.2.
- **Recover**: We create for each supply plan different alternatives based on the recovery strategies implemented by the functions `try_strategic_stock(), try_alternative_mode()` and `try_later_recovery()`. Afterward, we execute `add_rest_plan()` to ensure that the non-disrupted parts of the plan are modeled as we link these new plans to the order. In case of an external disruption, we implement the `try_alternative_suppliers()` to find alternative suppliers to provide the same intermediate products or materials, for the same time as the disrupted supplier. In case it is successful, we execute `allocate_supplier_product(), propagate_capacity()` to reflect the new capacities allocated to compensate for the disrupted supply.
- **Evaluate**: For each order, we create alternative plans. in order to evaluate the disruption as mentioned in subsection 5.2.3, we execute `get_plan_quantity()` to evaluate if the final quantity in the supply plan is equivalent to the pre-disruption quantities. Similarly, `get_plan_price()` and `get_latest_plan_time()` are to calculate the price and the delivery time of the alternative plans; The result of the previous methods can be aggregated per customer or for the overall orders.

D.4 Reproducibility

The current state of MARE allows the reproducibility of the resilience evaluation. We can examine the effect of different disruptions on the SC by modifying in the first step (`create_disruptions`) to model more or different disruptions. MARE's reproducibility supports the study of the effect of disruptions on another instance of the SC, i.e, new version of *src/resources/supplychain.ttl*. We can also extend the disruption model to include more properties enabling further DMP analysis, i.e, new version of *src/resources/disruption.owl*. For instance, we can include the SC stakeholders handling the disruption. The behavior of MARE can be reproduced with different recovery strategies and metrics evaluation.

APPENDIX E

KnowGraph-PM: A Knowledge-Graph-based Pricing Model for Semiconductor Supply Chains

We present the SPARQL queries translated from the competency questions in KnowGraph-PM section 6.1

Listing E.1: Query for CQ1: Get top 20 most profitable customers.

```
PREFIX : <http://www.example.org/LTBP#>.
SELECT ?customer(SUM(?price) as ?TotalRevenue)
WHERE {
  ?order    :wasPlacedBy ?customer.
  ?customer :hasPricePremium ?x.
  ?order :containsProduct ?product.
  ?order :hasRMPrice ?price.
}
GROUP BY ?customer
ORDER BY DESC (?TotalRevenue) LIMIT (20)
```

Listing E.2: Query for CQ2: Get LTBP occurrence-based customer ranking where the divisor is the count per customer types

```
PREFIX : <http://www.example.org/LTBP#>.
SELECT ?type (COUNT(?temp) as ?RMPossible)
(IF(?type = "Regular", ?RMPossible/10073,
IF(?type = "Key",?RMPossible/37462, ?RMPossible/2908)) as ?percentage)
FROM <http://infineon.com/LTBP/>
WHERE {
  ?order :wasPlacedBy ?customer.
  ?order :hasRequestedOrderLeadTime ?rolt.
  ?order :hasStandardDeliveryTime ?sdt.
  ?customer :hasType ?type.
BIND((?sdt*7) - ?rolt as ?temp)
FILTER(?temp > 0)
}
GROUP BY ?type
```

Listing E.3: Query for CQ3: Get Per customer class price premium estimation.

```
PREFIX : <http://www.example.org/LTBP#>.
SELECT ?type (MAX(?x) as ?MaxRM) (MIN(?x) as ?MinRM) (AVG(?x) as ?AverageRM)
WHERE {
  ?customer :hasPricePremium ?price .
  ?customer :hasType ?type .
}
GROUP BY ?type
```

Listing E.4: Query for CQ4: Get initial customer and product selection.

```
PREFIX : <http://www.example.org/LTBP#>.
SELECT ?product ?customer (SUM(?price) as ?TotalProductRevenue)
FROM <http://infineon.com/LTBP/>
WHERE {
  ?order :wasPlacedBy ?customer.
  ?customer :hasPricePremium ?pricepremium.
  ?order :containsProduct ?product.
  ?order :hasRMPrice ?price.
}
ORDER BY DESC ?customer ?product
```

APPENDIX F

An Ontology-based Approach for Preprocessing in Machine Learning for Packing Information

F.1 Competency Questions, SPARQL Queries and Results

We present the SPARQL queries translated from the competency questions in section 6.3

F.2 Feature Analysis

Figure F.1 and Figure F.2 compare the feature relevance between the ontology-based and “baseline” approaches. We obtain the presented graphs for the relevant features as an output of the Random Forest model. This set of features are a reference for the user to know which are the variables which are most commonly used by the model for the targets prediction.

We compare the graphs of the ontology-based and “baseline” approaches and we consult the domain experts to assess which set of features based on their knowledge were more representative. The experts concluded that the relevant features from the ontology-based approach are more representing. They especially remarked that the categorical feature “functional_packing” and its different categories (e.g., “functional_packing_tube_method” and “functional_packing_tray_method”) are in their opinion the most important features.

Likewise, the package category and location played an important role due to the similar products that are produced in each location. This representative set of features, which is enabled by the ontology approach, can further guide into the discovery and selection of relevant features. Likewise, having identified the relevant features, the user can deploy this information for further data cleaning tasks and creation of restrictions that improved the values and relations for those features considered as relevant.

Competency Question

```
SELECT DISTINCT ?property
WHERE {
  ?packing_info rdf:type :packing.
  {?packing_info ?property ?object.}
UNION
{
  ?subject ?p ?packing_info.
  ?subject owl:hasValue ?value.
  ?subject owl:onProperty ?property.
}
}
ORDER BY ?property
```

SPARQL Result	**property**
	has_creation_date
	has_devices_per_box
	has_devices_per_functional_packing
	has_functional_packings_per_box
	has_outer_box_dimension_unit
	has_outer_height
	has_outer_length
	has_outer_width

Table F.1: SPARQL query and results of competency question CQ1: "What are the properties of a packing info?".

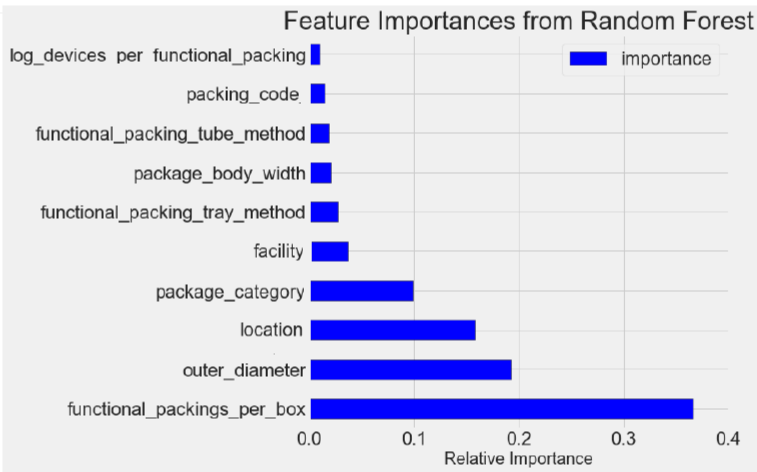


Figure F.1: Relevant features as an output of the Random Forest model to assess which set of features is more representative for the ontology-based approach.

Competency Question

```
SELECT DISTINCT ?sales_product  ?package_name
 ?packing_info
WHERE {
  ?sales_product rdfs:subClassOf :
      semiconductor_product.
  ?sales_product rdfs:subClassOf ?a.
  ?a owl:someValuesFrom ?package_name.
  ?a owl:onProperty ?property.
  ?package_name rdfs:subClassOf :package.
  ?package_name rdfs:subClassOf ?b.
  ?b owl:hasValue ?packing_info.
  ?b owl:onProperty ?property2.
  ?packing_info rdf:type :packing.
}
```

SPARQL Result

sales_product	package	package_info
ACIC7-2TN	PG-TO220-5-12	A66766-S1030-Z553-A0-74A9
ACIC7-2TN	PG-TO220-5-12	A66766-S1030-Z667-A0-74A9
ACIC8TN	PG-TO220-5-12	A66766-S1030-Z553-A0-74A9
BTS244Z-E3043	PG-TO220-5-12	A66766-S1030-Z667-A0-74A9

Table F.2: SPARQL query and results of competency question CQ2: "How is a packing info assigned to a product?".

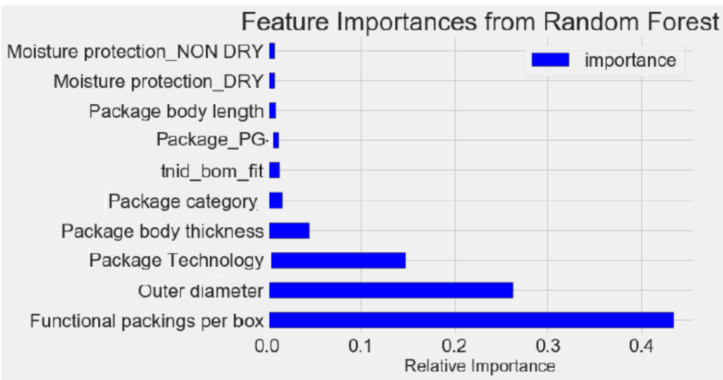


Figure F.2: Importance feature as an output of the Random Forest model in the “baseline” approach.

Competency Question

```
SELECT DISTINCT  ?d_product ?funct_packing
?c_product ?funct_packing2
WHERE{
{
  ?d_product rdfs:subClassOf :semiconductor_product.
  ?funct_packing rdfs:subClassOf :dice_packing.
  ?funct_packing rdfs:subClassOf ?a.
  ?a owl:allValuesFrom ?dice_product.
  ?a owl:onProperty :is_packing_of_dice.
}
UNION {
  ?c_product rdfs:subClassOf :semiconductor_product.
  ?funct_packing2 rdfs:subClassOf :component_packing.
  ?funct_packing2 rdfs:subClassOf ?b.
  ?b owl:allValuesFrom ?component_product.
  ?b owl:onProperty :is_packing_of_component.
}
}
```

SPARQL Result	?d_product	?funct_packing	?c_product	?funct_packing2
	dice_product	blister_tape		
	dice_product	wafer_sawn		
			component_product	ammo_pack
			component_product	lister_tray_method

Table F.3: SPARQL query and results of competency question CQ3: "How is a functional packing assigned to each product type?".

Curriculum Vitae

Personal Details

Name	Nour Ramzy
Date of Birth	03.08.1993
Email	nourhany1993@gmail.com
Family status	Single

Education

2007-2010	French Baccalaureate (High School), College de la Mere de Dieu, Cairo, Egypt
2011-2016	BSc in Communication and Computer Engineering, Faculty of Engineering, Cairo University, Cairo, Egypt
2016- 2019	MSc in Computer Hardware and Software Computer Engineering, University of Stuttgart
2019-2022	PhD Candidate at Gottfried Wilhelm Leibniz Universität Hannover and Infineon

Professional Experience

Jul 2012	Intern at Schlumberger, Cairo, Egypt
Jul 2013, 2014	Intern at IBM Egypt
Jun-Aug 2015	Intern at EME international- Mobile Development Company- Cairo, Egypt
May-Sept 2017	Research Assistant at Institut für Technische Informatik, University of Stuttgart , Stuttgart, Germany
Apr- Aug 2018	Intern at Infineon Technologies Munich, Germany
Aug 2018 - Current	Team Lead for the Semantic Web team at Infineon
	Project Management for European Projects: Work package and deliverables
Aug 2019- Current	PhD Candidate and Team Lead for the Semantic Web team at Infineon

Technical Skills

Semantic Web	Semantic Data Integration: ontology modeling , data mapping, querying
	Semantic Web stack: XML, URI, RDF, OWL, RDFS, SPARQL
Coding Languages	Java, C, C++, C#, matlab
Database Systems	SQL, Oracle, mongoDB, Data Warehousing and OLAP
Web Development	HTML, XML, javascript, CSS
Others	Languages and compilers, Operating Systems management, Object Oriented Programming, Artificial Intelligence basic techniques

Languages

Arabic	Mother tongue
English	Fluent
French	Fluent
German	Basic
Spanish	Basic