DOKTORI (PhD) ÉRTEKEZÉS

Nagy László

Pannon Egyetem
2023

PANNON EGYETEM

DOKTORI (PhD) ÉRTEKEZÉS

Ipar 4.0 és 5.0 megoldások fejlesztése ontológiák alapján - modellezés és optimalizálás

---

Szerző:
NAGY László

Konzulensek:
Prof. Dr. habil. ABONYI János
Dr. RUPPERT Tamás

DOI:10.18136/PE.2023.845

Értekezés doktori (PhD) fokozat elnyerése érdekében
*a **Pannon Egyetem***

Vegyészmérnöki- és Anyagtudományok
Doktori Iskolájához tartozóan

Folyamatmérnöki Intézeti Tanszék

Pannon Egyetem
2023

Ipar 4.0 és 5.0 megoldások fejlesztése ontológiák alapján - modellezés és optimalizálás

Az értekezés doktori (PhD) fokozat elnyerése érdekében készült a Pannon Egyetem
Vegyészmérnöki- és Anyagtudományok Doktori Iskolája keretében

Bio-, környezet- és vegyészmérnöki tudományok tudományágban

Írta: Nagy László

Témavezetők: Dr. Abonyi János, Dr. Ruppert Tamás

Elfogadásra javaslom (igen / nem)

………………………. ……………………….
(témavezetők)

Az értekezést bírálóként elfogadásra javaslom:

Bíráló neve: Dr. Kiss Attila igen /nem

……………………….
(bíráló)

Bíráló neve: Dr. Gyulai Dávid igen /nem

……………………….
(bíráló)

A jelölt az értekezés nyilvános vitáján …..........%-ot ért el.

Veszprém, ……………..………………………
(a Bíráló Bizottság elnöke)

A doktori (PhD) oklevél minősítése…................................

Veszprém,

………..…………………..….
(az EDHT elnöke)

UNIVERSITY OF PANNONIA

DOCTORAL (PhD)THESIS

Ontology-based development of Industry 4.0 & 5.0 solutions - modeling and optimization

---

Supervisors:
Prof. Dr. habil. János ABONYI
Dr. Tamás RUPPERT

Author:
László NAGY

A thesis submitted in fulfilment of the requirements
for the degree of Doctor of Philosophy

in the

Doctoral School in Chemical Engineering and Material Sciences
of University of Pannonia

Department of Process Engineering

University of Pannonia
2023

Ha tudnánk, mit csinálunk, akkor nem neveznénk kutatásnak, igaz?

Albert Einstein

If we knew what it is we were doing, it would not be called research.
Would it?

Albert Einstein

PANNON EGYETEM

Kivonat

Mérnöki Kar
Folyamatmérnöki Intézeti Tanszék

Philosophiæ Doctor

Ipar 4.0 és 5.0 megoldások fejlesztése ontológiák alapján - modellezés és optimalizálás

írta: Nagy László

A hatékony információkezelés kritikus fontosságú a gyártási folyamatok fejlesztéséhez, különösen az intelligens gyártás korában, ami a kölcsönhatásban lévő elemek kritikus halmazainak megfelelő modellezését és szisztematikus elemzését igényli. A kutatás célja, bemutatni egy ontológiamodell-alapú keretrendszert az Ipar 4.0 megoldások fejlesztéséhez, valamint a gyártási adatok gráf alapú optimalizálására szolgáló technikákat. A szemantikus technológiák átfogó áttekintése rávilágít arra, hogy a gráftechnológiák integrálása a meglévő ipari szabványokba, tervezési és végrehajtási rendszerekbe hatékony adatfeldolgozást és elemzést biztosíthat. További vizsgált probléma az Ipar 5.0 megoldások tervezése, amely a gyártási folyamatnak, az operátorok képzettségeinek és állapotainak, valamint az intelligens térben elhelyezett szenzoroknak a kollaboratív munka egyidejű megfigyeléséhez szükséges problémaspecifikus leírását igényli. A hipotézis szerint az ontológia-alapú adatok hatékonyan reprezentálják a vállalati és gyártási adathalmazokat, továbbá az így kapott gráfok centralitásának és modularitásának vizsgálata képes támogatni a kollaborációs és interakciós sémák kialakítását, valamint a gyártási cellák tervezését. A dolgozat eredményei két részben, a modellezés, illetve az optimalizálás témakörében kerülnek bemutatásra. A szemantikus modellezési rész áttekintést nyújt az Ipar 4.0 és 5.0 alkalmazások létrehozásában felhasználható ontológiákról és tudásgráfokról, továbbá két részletes alkalmazást mutat be egy reprodukálható ipari esettanulmányon. Az optimalizálási rész a hálózattudomány alapú folyamatoptimalizálásra összpontosít, és különböző részletes alkalmazásokat mutat be, például gráf alapú analitikát, gyártósor kiegyenlítést, valamint közösség detektálást.

UNIVERSITY OF PANNONIA

Abstract

Faculty of Engineering
Department of Process Engineering

Doctor of Philosophy

Ontology-based development of Industry 4.0 & 5.0 solutions - modeling and optimization

by László Nagy

Effective information management is critical for developing manufacturing processes, especially in the era of smart manufacturing, which requires adequate modeling and systematic analysis of the critical sets of interacting elements. This research aims to present an ontology model-based framework for developing Industry 4.0 solutions and a collection of techniques for graph-based optimization of manufacturing data. An extensive overview of semantic technologies is provided, highlighting that integrating graph technologies into existing industrial standards, planning, and execution systems can provide efficient data processing and analysis. An additional investigated problem is the design of Industry 5.0 solutions, which require a problem-specific description of the production process, the skills and states of the operators, as well as of the sensors placed in the intelligent space for the simultaneous monitoring of the collaborative work. The hypothesis is that ontology-based data can efficiently represent enterprise and manufacturing datasets, moreover, studying the centrality and modularity of the resultant graph can support the formation of collaboration and interaction schemes and the design of manufacturing cells. The contributions of the thesis are presented in two parts, modeling and optimization. The semantic modeling part provides an overview of ontologies and knowledge graphs that can be utilized in creating Industry 4.0 & 5.0 applications, furthermore presents two detailed applications on a reproducible industrial case study. The optimization part of the work focuses on the network science-based process optimization and presents various detailed applications, such as graph-based analytics, assembly line balancing and community detection.

PANNONISCHE UNIVERSITÄT

Auszug

Fakultät für Ingenieurwissenschaften
Abteilung für Verfahrenstechnik

Doktor der Philosophie

Ontologiebasierte Entwicklung von Industrie 4.0 und 5.0 Lösungen - Modellierung und Optimierung

von László NAGY

Ein effektives Informationsmanagement ist für die Entwicklung von Fertigungsprozessen von entscheidender Bedeutung, insbesondere im Zeitalter der intelligenten Fertigung, die eine angemessene Modellierung und systematische Analyse der kritischen Gruppen von interagierenden Elementen erfordert. Diese Forschung zielt darauf ab, einen auf Ontologiemodellen basierenden Rahmen für die Entwicklung von Industrie 4.0-Lösungen und eine Sammlung von Techniken für die graphenbasierte Optimierung von Fertigungsdaten zu präsentieren. Es wird ein umfassender Überblick über semantische Technologien gegeben und hervorgehoben, dass die Integration von Graphentechnologien in bestehende Industriestandards, Planungs- und Ausführungssysteme eine effiziente Datenverarbeitung und -analyse ermöglichen kann. Ein weiteres untersuchtes Problem ist die Gestaltung von Industrie 5.0-Lösungen, die eine problemspezifische Beschreibung des Produktionsprozesses, der Fähigkeiten und Zustände der Bediener sowie der im intelligenten Raum platzierten Sensoren zur gleichzeitigen Überwachung der kooperativen Arbeit erfordern. Die Hypothese ist, dass ontologiebasierte Daten Unternehmens- und Fertigungsdatensätze effizient darstellen können. Darüber hinaus kann die Untersuchung der Zentralität und Modularität des resultierenden Graphen die Bildung von Kollaborations- und Interaktionsschemata und die Gestaltung von Fertigungszellen unterstützen. Die Beiträge dieser Arbeit werden in zwei Teilen vorgestellt: Modellierung und Optimierung. Der semantische Modellierungsteil bietet einen Überblick über Ontologien und Wissensgraphen, die bei der Erstellung von Industrie 4.0 und 5.0-Anwendungen genutzt werden können, und stellt darüber hinaus zwei detaillierte Anwendungen anhand einer reproduzierbaren industriellen Fallstudie vor. Der Optimierungsteil der Arbeit konzentriert sich auf die netzwerkwissenschaftlich basierte Prozessoptimierung und stellt verschiedene detaillierte Anwendungen vor, wie graphbasierte Analytik, Fließbandbalancierung ausgleichen und Community-Erkennung.

Acknowledgements

Above all, I am very grateful to my supervisors. Prof. Dr. habil. Janos Abonyi, your support, and encouragement throughout the years were essential to reach this point. You helped me a lot, highlighting the "tiny correlations in the big picture" and motivating me to dive deep into new research topics. Thank you again for the opportunity to spend a week at the University of Catania at the beginning of my studies, which gave me a boost and really could grab my attention in applied network science.

Dr. Tamás Ruppert, first of all, thank you for drawing my attention to this career path and supporting me throughout this journey. I highly appreciate your easygoing attitude, which made my doctoral training program a lot easier to be successful in working remotely. I am also grateful for the ETFA conference spent together in Stuttgart, which was a great experience.

Family and Friends. I know it can be challenging to tolerate my grumpiness if I am stressed by work, thank you for not letting me down. I am honored to be surrounded by such good-natured people. Your positivity and encouragement helped me a lot to deal with the challenges. Thank you!

Dedicated to my Family and Friends. Because without them, I would not have been able to accomplish this journey.

Contents

Chapter A

Introduction and motivation of the thesis

This PhD thesis is divided into two parts. Part I. introduces the field of semantic-based modeling, using ontologies and knowledge graphs and shows application examples in Chapter I.2 and I.3. Part II. discusses the network science-based process optimization, where advanced manufacturing analytics is applied, using graphs, and presents several methods in Chapters II.2-II.4. This Chapter aims to provide the theoretical background and general problem statement, covering both parts of the thesis.

First, Section A.1 summarizes the studied fields of engineering, network science, and emerging technologies that form the research problem. Subsection A.1.1 gives a background to ontologies, standards and semantic networks , while in subsection A.1.2, the Industry 5.0 related field is described. Finally, subsection A.1.3 gives the list of the problem statement. Additionally, Section A.2 presents a framework that combines semantic-based modeling and network science-based process optimization to develop Industry 4.0 & 5.0 solutions. Finally, the main research questions of this PhD thesis are stated in Section A.3.

A.1 Introduction of the research topics - problem statement

This section describes the technology topics that lead to the research questions. First, subsection A.1.1 presents the field of related industry standards and ontology-based modeling, then subsection A.1.2 highlights the challenges of Industry 5.0 and presents the human-centric approach in smart manufacturing. Finally, subsection A.1.3 summarizes the problem statement.

A.1.1 Standards and ontology-based modeling of manufacturing

This subsection first gives the background to understanding ontology-based modeling and its potentials. The concept of Industry 4.0 has already significantly influenced how production and assembly lines are designed [1] and managed [2]. As Internet of Things-based products and processes are rapidly developing in the industry, there is a need for solutions that can support their fast and cost-effective implementation. There is a need for further standardization to achieve more flexible connectivity, interoperability, and fast application-oriented development; furthermore, advanced model-based control and optimization functions require a better understanding of sensory and process data [3].

Managing information and data from production systems is critical for digital transformation, especially in Industry 4.0 applications where the horizontal and vertical integration of systems require more efficient data processing [4]. More efforts have been made to standardize this area, such as the ANSI/ISA-95[1] international standard or the RAMI 4.0[2] [5]. Furthermore, there are ongoing studies in the field of different methodologies and data structurization that aim to support production-related decision-making processes [6] or create models without simulation software-specific knowledge [7]. For a similar purpose, process mining solutions were also developed to discover, analyse, and improve business processes based on event logs of information systems [8, 9].

---

[1] International Society of Automation

[2] Reference Architectural Model Industrie 4.0

To efficiently manage big data in a product life cycle or just on the shop floor, suitable methods such as digital twins are required. Although a digital twin is a digital replica (model) of a physical system and information flow between the elements of a complex system, they have various realisations. A digital twin system may have different functionalities, such as data collection, data processing, simulation, auto decision, synchronization and visualization [10]. Moreover, it highlighted that ontologies and semantic approaches also play an important role in developing digital twins [11]. The main objectives of a paradigm for architecting digital twins for manufacturing processes are to ensure the following factors: Modularity, Scalability, Reusability, Interoperability and Composability [12]. One approach uses the IEC 61499 standard and includes multilayered, multi-levelled (inspired by the ISA-95 standard) and multi-perspective concepts for building a digital twin for manufacturing processes. Furthermore, it presents an ontology-based implementation, the Digital Twin Architecture Ontology Model (DATOM) [12].

Capturing knowledge is demanded in digital formats concerning different aspects of the industry such as process planning, production, or design is increasing, as the variety and complexity of product lifecycle applications have risen. It is hypothesized that knowledge graphs, semantic web technologies and multi-agent systems will be the driving forces to form data into knowledge and evolve how processes are automated using interoperable data [13]. Furthermore, establishing procedures to facilitate the structured and objective representation or communication of domain-specific knowledge is essential in terms of smart manufacturing.

The Semantic Web stands for an extension of the World Wide Web with standards aiming to make the internet data machine-readable. It involves publishing in languages specifically designed for data, such as Extensible Markup Language (XML), Resource Description Framework (RDF) and Web Ontology Language (OWL). An ontology can be determined as a graph-based data model that manages how entities (individuals) are grouped into categories (classes) and which appear on the most fundamental level. Additionally, ontologies can describe real-world phenomena and their relationships among each other in a machine-readable way by using formal elements, such as instances, rules, relationships and axioms [14]. A knowledge graph is a highly flexible non-SQL (Structured Query Language) database representing data as “knowledge” through a graph-like structure of nodes and edges. The nodes that refer to the knowledge are often defined in an ontology, the concepts that describe the domain. They can be traversed semantically using domain knowledge.

Additionally, ontology models can facilitate contextualizing the KPIs (Key Performance Indicator) [15], detecting indirect effects or influences, and analysing relationships within a complex network [16]. Furthermore, these can support the thematic visualization of KPIs, the creation of dashboards [17], and the aggregation of KPI related data [18]. Once the relationship with the decision variable can be described in this form, responsive development and optimization become possibble.

Based on the literature, it can be stated that in the era of Industry 4.0, efficient data management systems are required, and semantic technologies combined with the existing industrial standards can offer a solution. Although production models, semantic technologies, and graph representation can offer the technical background, with the emerging Industry 5.0 and human-centric approach, additional aspects must be addressed, as presented in the following subsection.

A.1.2 Human-centric and collaborative approach - Challenges of Industry 5.0

This subsection aims to give an introduction to the human-centric approach of modern industry. A strong necessity to increase productivity while not removing human workers from the manufacturing industry creates challenges for the global economy and developers of MES (Manufacturing Execution System) or ERP (Enterprise Resource Planning) systems [19], where the operator is still not sufficiently integrated. The main aspects of Industry 4.0 aim to extensive digitalisation, while in an Industry 5.0 environment, the goal is to integrate innovative technologies with human actors, which can be stated as a more value-driven than technology-driven approach [20]. Industry 4.0 focuses less on the original principles of social fairness and sustainability and more on digitalisation and artificial intelligence-driven technologies to increase flexibility and efficiency [21]. Industry 5.0 complements and extends the main features of Industry 4.0. At the same time, it provides a different focus and highlights the importance of research and innovation to support industry in its long-term service to humanity [21]. Additionally the research interest is emerging in aspects of industrial humanization [22], sustainability and resilience [23]. Figure A.1 represents the main goals of the Industry 5.0 concept, which was not part of Industry 4.0, as the production should be not only digitalized, but also resilient, sustainable and human-centric.

Figure A.1: The main pillars of Industry 5.0 [20]

From a human-centric point of view, the concept of Industry 5.0 [24] is considered, where robots are intertwined with the human brain and work as a collaborator instead of as a competitor. Integrating all parts of production, business processes as well as Information and Communications Technologies facilitates the formation of a complete digital copy of production as a digital twin. Therefore, a reflection of all the fundamental physical processes in a virtual production model is achieved, nevertheless, the results of digital modeling can provide feedback and control real production processes, which are integral parts of the concept of Industry 5.0 [25]. As a result, human intelligence is a dominant and decisive factor in intelligent manufacturing, which is consistent with the concept of Human-Cyber-Physical Systems (H-CPSs) [26].

Future intelligent factory ensures the synergy of the skills of machines (such as robots) and humans to increase productivity and maintains healthy, safe and sustainable working conditions [27]. One of the biggest challenges of modern manufacturing is to create an adequate human-machine relationship in complex human-machine systems, especially when a strong synergy between the capabilities of machines and humans is needed. Personalized work instruction systems can facilitate human-machine interaction, utilizing dynamic knowledge profiling and importing [28]. Additionally, direct collaboration or task sharing within the same working area requires connecting machines even more closely to humans [29].

In a so-called human-in-the-loop smart manufacturing concept, digitalisation aims to facilitate relationships between humans and manufacturing sites [30]. Similarly, in a human-centric smart manufacturing concept, the goal is to develop a H-CPS [31]. The human influence on CPS (Cyber-Physical System) plays a dominant role in the formation and development of CPS, e.g., the cognitive skills are taken into account in interface design [32]. Therefore, human intelligence is a dominant and decisive factor in intelligent manufacturing, consistent with the concept of H-CPS [26].

The human-centric manufacturing aims that the industry should place the well-being of shop floor workers at the center of manufacturing processes instead of being system-centric. Practice should ultimately address human needs defined in an Industrial Human Needs Pyramid – from safety and health to the highest level of esteem and self-actualization. The five levels of industrial human needs, and the five steps between them, are the followings: safety - coexistence, health - cooperation, belonging - collaboration, esteem - compassion, self-actualization - coevolution, based on Ref. [33]. The main aspects of Industry 4.0 aim to reach extensive digitalisation, while in an Industry 5.0 environment, the goal is to integrate innovative technologies with human actors or can be regarded as more value-driven than technology-driven approach [34, 35]. From the motivations mentioned above, the concept of Industry 5.0 is considered [24], where robots are intertwined with the human brain and work as collaborators.

The so-called industrial immersive technologies (IIT) summarize technical solutions which play a key role in forming the new industrial revolution and the H-CPS for complex manufacturing processes. A wide range of papers and patents are available in this area, which also serves as a source of a domain ontology for IIT in Ref. [36]. The overview of technology specifications divides these tools into four main groups: brain-machine interfaces, virtual reality, augmented reality and industrial engineering [36].

A new trend in the research and development of human factors as well as the stochastic nature of humans during manufacturing processes is the Operator 4.0 concept, which proposes eight different types of how workers on the shop floor can be supported [37]. A workforce is one of the most critical manufacturing resources as well as the most agile and flexible, therefore, the improvement of human operator resilience can also make manufacturing systems more resilient, which is discussed in the Resilient Operator 5.0 concept [38]. The central element of these solutions is the integrated monitoring of the activities of the operators and the manufacturing system. The Resilient Operator 5.0 concept is defined as a competent and skilled shop-floor worker using human creativity, ingenuity and innovation, aided by information and technology to overcome difficulties or obstacles. At the same time, it is also aimed to develop additional and cost-effective solutions by the stakeholders to ensure long-term sustainability and workforce well-being in manufacturing while facing unexpected conditions [38]. The development of the enabling technologies of the Operator 5.0 concept requires a wide field of research. Various frameworks integrating digital technologies such as Extended Reality, Big Data Analytics, Artificial Intelligence, and Digital Twins must be standardized for industrial usage. A survey set three main characteristics as assisting parameters for this goal: human-centricity, (social) sustainability, and resilience [39].

Another research topic, namely the Intelligent Factory Space (IFS) concept, represents a framework for interaction between humans and an automated system (digital factory) for which three key features are proposed: Observing, Learning and Communication [29]. The IFS is composed of multiple layers (representing different services for the human user) and many modular components, which can be extended to meet the requirements of users. The IFS relies on industrial standards to communicate with existing machines while using novel two-way communication possibilities to feedback to the human user [29]. A further approach, which needs to take into account is the Smart Factory concept [40]. It aims to apply technologies that lead to adaptive and flexible manufacturing such as IIoT (Industrial Internet of Things) devices or cloud services [41].

Based on this introduction to the human-centric approach, it can be stated, that there is a high demand on integrating human factors in CPSs and on developing adequate methods to facilitate collaboration and ergonomics of the shop floor workers.

After the presented three main fields, in the following subsection the problem statement of the present thesis is summarized.

A.1.3 Problem statement of the thesis

Based on the evaluation of the above-presented literature study, defining the research gaps, and development tasks, this subsection summarizes the problem statement of this thesis:

- Horizontal and vertical integration is needed in modern industry applications for interoperability and standardization.
- Integrate semantic technologies into ERP and MES systems to facilitate data access and contextualization using graph-based representation.
- Develop optimization methods for Industry 4.0 and 5.0 concepts, while adapting network-based process models to industry standards.
- A large amount of data is needed to be processed in a way where interoperability and re-usability factors are also satisfied.
- Integrate human factors into CPSs and develop adequate methods, based on the Industry 5.0 concept, to facilitate collaboration and support the shop floor workers.

After the research background of the studied field, the following subsection introduces the proposed framework to handle the formulated problems.

A.2 Proposed framework for ontology-based development of Industry 4.0 & 5.0 solutions

This section describes a methodology suitable for developing production-related ontologies and knowledge graphs and integrating graph-based models with network science-based optimization methods.

Figure A.2 serves as a graphical abstract for the entire thesis. The blue elements represent Part I., Semantic modeling, and the orange elements of the figure show the Part II., Advanced manufacturing analytics topic of this thesis. The main contribution of this framework is that the value of the information is increasing, thanks to semantic data and data enrichment. The data collection and the graph database parts of the framework are outside of the scope of the thesis and not discussed in detail, as this work focuses on modeling and optimizing complex systems.

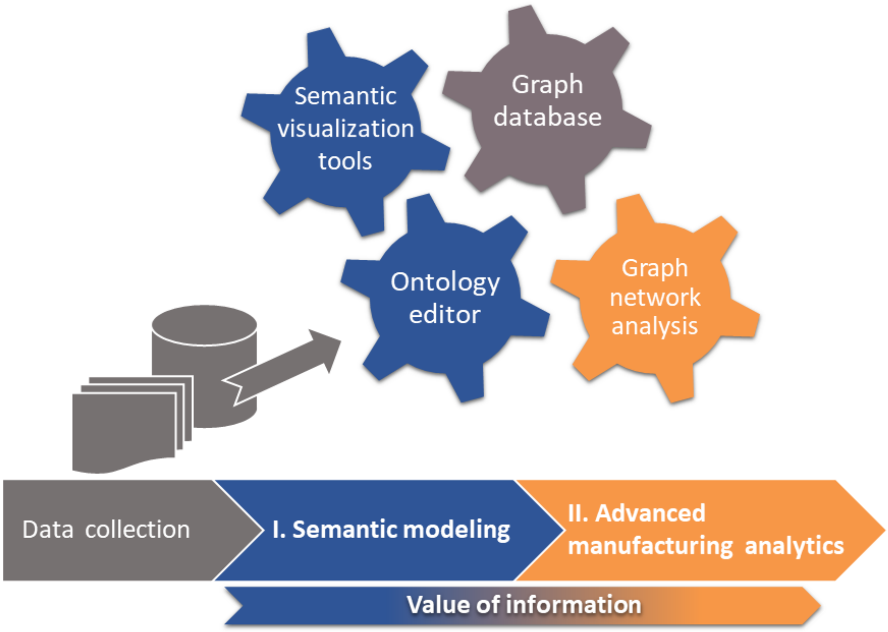


FIGURE A.2: Graphical abstract of the PhD thesis - Main elements of the proposed framework

The following list gives a more in-depth description of the graphical abstract that describes the proposed framework:

- Data collection
  - The complex manufacturing processes of an Industry 4.0 environment create a large amount of data, using variety of IoT (Internet of Things) devices and sensors.
  - The method performs pre-processing on the raw data and transforms production data into ontology-based databases.
  - The technology solutions and methods for data collection are not in the scope of this thesis, therefore, this topic is not discussed in detail.

I. Semantic modeling - Part I.
  - Adequately structured and contextualized production data is required.
  - The semantic modeling method establishes the basic structural network of the production process and includes interactions between groups or classes.
  - This solution determines descriptive and influential factors of the system as cost parameters, requirements or optimizable elements.
  - The development of the desired ontology, using appropriate software tools is also required, additionally, the databases of production processes have to be connected with the developed ontology.

- Graph database
  - Graph databases are required to serve as a bridge within Part I. and II. and provide accessible graph data for analysis and optimization algorithms.
  - Adequate procedures convert the semantic web-based data into a graph database.

II. Advanced manufacturing analytics - Part II.

  - Using graph databases, a variety of process analysis and optimization methods are accessible, which aim to meet the needs of Industry 4.0 and 5.0.
  - Generate labelled multilayer networks from the ontology models, and graph databases.
  - Create visualization of the knowledge graph with normal, directed or hypergraph.
  - Analyse the previously defined descriptive and influential factors with data queries.
  - Utilize clustering and community detection on the aggregated graph data.
  - Use the enriched information to solve assembly line balancing problem, optimize allocations, or support cell formation and layout design.

An essential step of the proposed methodology (also highlighted in Figure A.2) is the application of visualization tools that can be beneficial during ontology modeling and advanced manufacturing analytics phases as well. Appropriate visualization of the created ontology with graph diagrams can support the development process or provide additional internal information about the manufacturing procedure.

Additionally, Figure A.3 highlights the main engineering challenges and the corresponding analysis tasks, such as the assembly line balancing problem can be solved with multilayer analysis, the community detection algorithms can be utilized for allocation optimization, and the evaluation of centrality metrics can help to handle the complexity of the network.

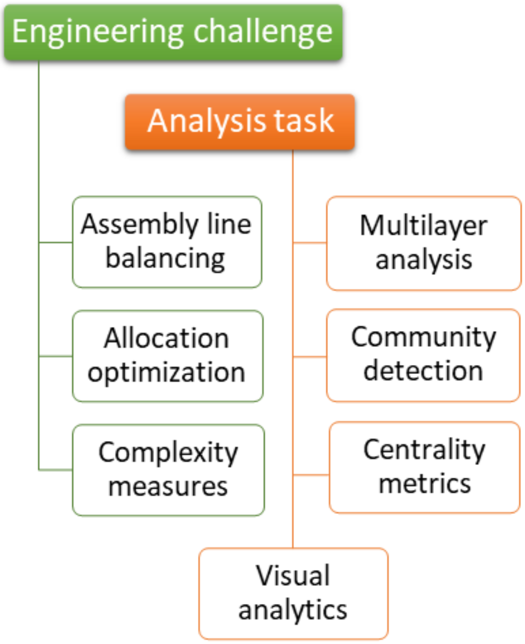


FIGURE A.3: Engineering challenges and tasks to solve in the present thesis

After presenting the research background, and the problem statement, the following subsection describes the main research questions of this thesis.

A.3 Research questions and thesis outline

Based on the presented three fields of research and the proposed development framework, this subsection states the research questions of the thesis.

Firstly, as a graphical summary, Figure A.4 represents the main stages of this PhD thesis, where the top aim is to facilitate *production optimization* and develop *Industry 4.0 and 5.0 solutions*. Regarding the two parts of this work, the elements are colored differently as connecting to *modeling* (blue) or *optimization* (orange).

The two main pillars of the thesis are: performing *process modeling* to create *network-based representation* (from a complex production or Industry 4.0 application), using *ontologies and standards*; additionally satisfying the *goals and limitations* of the production process, using *multilayer analysis* for *optimized allocation*, and develop *optimization algorithms* for *multi-objective production optimization*.

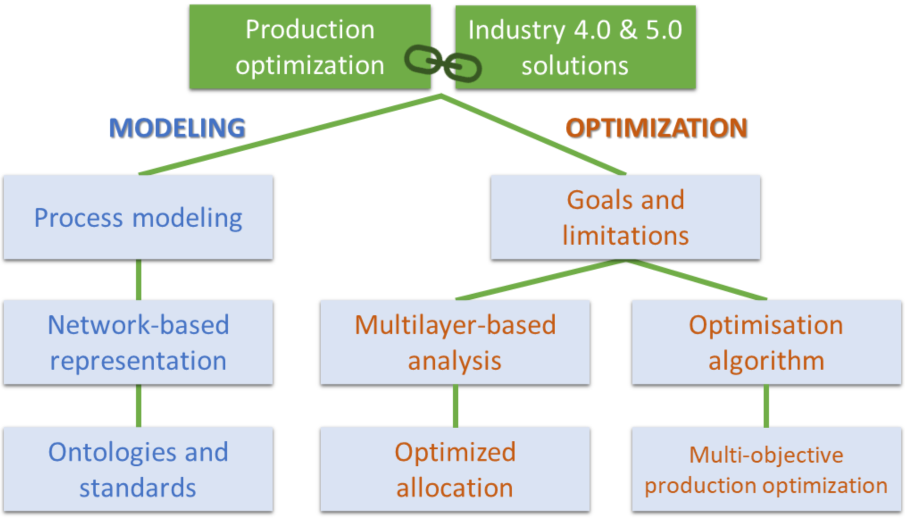


FIGURE A.4: The principal investigated field of the thesis in modeling and optimization categories, aiming to solve production optimization and create Industry 4.0 & 5.0 solutions

The main research questions of the present PhD thesis based on Section A.1 are the followings:

- *How semantic technologies and ontologies can be applied in the modern industry?*

  The question relies on whether the ontology-based system representation can serve as a basis for an efficient, structured analysis of production processes. Chapter I.2 describes an industry related example of semantic modeling and data query analysis, presented on a wire harness assembly-based case study.

- *How can knowledge graphs be applied to support human-centric manufacturing?*

  The main idea is that human-centric manufacturing can be effectively supported with knowledge graphs and ontology-based models. Chapter I.3 proposes the Human-Centric Knowledge Graph framework adapting ontologies and standards, that can model the operator-related factors, such as monitoring movements, work conditions, or collaboration with robots. Graph-based data query, visualization, and query analytics are also presented using an industrial case study.

- *Is it possible to integrate graph-based process modeling with multi-objective optimization, to solve the assembly line balancing problem?*

  The proposed approach is based on that the assembly line balancing problem can be solved with multi-objective allocation, combining an optimization algorithm with graph models for detecting complex relationships in production systems. Chapter II.2 introduces an optimization method that combines the analytic hierarchy process and the multilayer network-based production representation to solve the assembly line balancing problem.

- *How can community detection and clustering be more efficient in the case of large, graph-based datasets?*

  It is assumed that knowledge graphs store a large amount of data, therefore, clustering the semantic data requires adequate algorithms. The proposal is based on combining graph-based models with efficient segmentation and optimization algorithms, a toolbox can be developed to support Industry 4.0 solutions. Chapter II.3 presents how communities in complex networks can be detected by integrating barycentric serialization-based co-clustering and bottom-up segmentation algorithms.

- *Can hypergraph-based models be utilized for the analysis of collaborative manufacturing?*

  Hypergraphs provide an effective tool to describe complex production processes and analyse cases of supportive or simultaneous human-machine collaboration. Chapter II.4 describes a hypergraph-based analysis method that can be applied in collaborative manufacturing. The description of manufacturing systems includes the skills and states of the operators, as well as the sensors placed in the intelligent space for the monitoring of the collaborative work. Moreover, studying the centrality and modularity metrics of the resultant network can support human-machine collaboration, operator well-being, and ergonomics factors.

Concerning that some of the above-listed questions correspond to modeling and some to optimization, the following part of the work is divided into two parts.

Both parts of the present thesis start with an introduction chapter to the discussed topic with an overview of the related literature. Then, the discussion of the applied methodology and tools are followed by the description of the applied dataset and the analytical results. The chapters discussing the research works are followed by a conclusion section discussing the general ideas and results. Finally, the thesis ends with the findings, containing the contributions to the ontology-based development of Industry 4.0 solutions. The discussion of the wire harness assembly process, used as a case study in several chapters throughout the thesis, is presented in the Appendix.

Part I.

Semantic modeling - ontologies and knowledge graphs

This part discusses the semantic-based modeling, using ontologies and knowledge graphs, and presents two detailed application aspects in Chapters I.2 and I.3, using industry related case studies.

First, Chapter I.1 introduces the theoretical and research background of semantic modeling. Chapter I.2 presents a case study about developing an industry-specific knowledge graph. Furthermore describes, how to create and evaluate data queries with ontologies. In Chapter I.3 a more complex knowledge graph development is presented, which aims to support Industry 5.0.

Chapter I.1

Introduction to the industrial application of semantic technologies

Adequate information management is critical for the development of manufacturing processes. Therefore, this chapter aims to provide a systematic overview of ontologies that can be utilized in building Industry 4.0 applications and highlights that ontologies are suitable for manufacturing management. Additionally, industry-related standards and other models are also discussed.

The contents of this introduction section are the followings:

- First, Section I.1.1 presents the main features of semantic modeling technologies and the research trends.
- Section I.1.2 serves as an overview of the relevant standards and modeling methods that are used in industrial practice such as ISA-95.
- The semantic & syntax elements and the most commonly used ontologies in the manufacturing industry are discussed in Section I.1.3.
- In Section I.1.4 summarizes the product, process, resource modeling aspects.
- Section I.1.5 presents the field of semantic technologies and metrics to describe and support the operator in a Industry 5.0 environment.
    - First, in subsection I.1.5.1 the topic of human activity recognition is presented.

  - Subsection I.1.5.2 gives some examples of ontology-based modeling and support systems for ergonomics and collaboration.
  - In subsection I.1.5.3, the field of quality metrics to evaluate human-machine interactions is discussed.

- Additionally, examples of ontology-based solutions in the industry are presented in Section I.1.6.

- Finally, section I.1.7 provides a brief comparison of ontology-based data processing, with traditional relational databases, and summarizes the difficulties of industrial application.

I.1.1 Ontologies and semantic models in general

Semantic data-based modeling structures the data in a specific logical way [42]. Ontology models also contain semantic information to provide a basic meaning of the data and describe their internal relationships [43]. Knowledge graph models provide a framework for data integration, processing, analytics, and sharing as a collection of interlinked descriptions of entities – objects, events or concepts [44, 45].

Figure I.1.1 shows the emerging trend of research papers related to ontologies. The publication data have been gathered from Scopus, filtering to research and review articles, without a limitation to the field of the journal, as ontologies and knowledge graphs are applied in multidisciplinary fields. As can be seen, the technology appeared around 2002 and the rapidly increasing number of publications in the topic Knowledge Graphs confirm its success and wide applicability range.

Because of the importance of horizontal and vertical integration in Industry 4.0, ontologies are being used in production systems to share information over an increasingly wide range [5]. Manufacturing companies are faced with many information-sharing tasks such as B2M, M2M, and B2B (communication channels between business (B) and/or machine (M) units) [46, 47]. The information has to be transferred between information systems, optimization methods, or digital twin simulators [48]. Due to the growing demand, the previously proposed developments for ISA-95, ISA-88, AutomationML (Automation Markup Language) and B2MML (Business To Manufacturing Markup Language) industry standards as

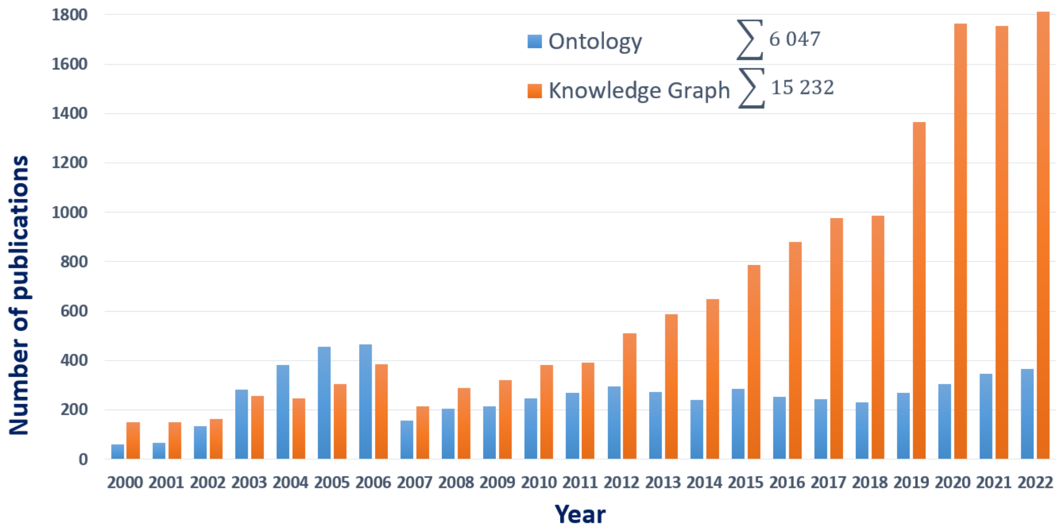


FIGURE I.1.1: Number of publications since 2000 in the topic of *Ontology* and *Knowledge Graph* - based on Scopus data

well as frameworks have intensified [5], moreover, these are based on knowledge graphs or ontologies.

During the fourth industrial revolution, new methods emerged to deal with this problem, and it can be stated that ontology modeling [49] and knowledge representation are part of the future trends [50], which can be presented using knowledge graphs. A knowledge graph is a programmatic method that subject-matter experts use to model a knowledge domain using data interlinking and machine learning algorithms. Its tasks refer to removing noise, inferring missing information and determining which facts should be included in a knowledge graph [51]. Also, new collaboration groups are forming as the *Industrial Ontologies Foundry* (IOF) [52, 53], or the European H2020 project *OntoCommons: Ontology-driven data for industry commons* [54] in order to standardization and to support the industry with advanced data interoperability, using reference ontologies. Another problem is the need to improve the efficiency of assembly processes in the manufacturing industry, where Computer-Aided Process Planning (CAPP) is gaining importance, which aims to identify appropriate resources while minimizing the total assembly time [55]. This is similar to the emerging importance of resource allocation and process planning in modern industry [56].

Ontologies and knowledge graphs are both used to represent knowledge and information in a structured way, but there are some differences between them. While ontologies typically consist of a hierarchy of concepts and relationships between them, often defined using a formal language like OWL or RDF. Knowledge graphs, on the other hand, allow for more flexible modeling, which can accommodate diverse types of data and relationships, including instances and attributes, often represented using complex graph databases. Furthermore, KGs can handle larger and more complex datasets, enabling better integration of disparate data sources, using more advanced reasoning and analysis, including natural language processing and machine learning, due to their graph database structure. Developing a knowledge graph instead of an ontology may require additional work in several aspects. As knowledge graphs can incorporate diverse types of data and relationships, there may be a greater need for data cleaning and extraction to ensure consistency and quality across the graph, so may require a more complex data integration and mapping between diverse data sources. Additionally, since knowledge graphs can support more advanced reasoning and analysis, there may be a need for additional expertise in machine learning, natural language processing, and other areas to fully leverage the capabilities of the graph [57]. In summary, a knowledge graph can provide a more detailed and complete representation of concepts and relationship, can support more complex queries, can be used for more advanced analytics and reasoning, or can provide better entity resolution, which means identifying and linking different instances of the same entity across different data sources, which can be particularly useful for large and complex datasets.

Studies of Gartner *Hype Cycle for Emerging Technologies, 2020* [58, 59] predict that *Ontologies and Graphs* as technological solutions are going to be available in two to five years. However, those are classified under the *Trough of Disillusionment* section, which means these technologies require special precautions to be applied effectively. Ontology modeling can be used for BPR (Business Process Re-engineering) or the development of control systems. Furthermore, it has a great importance as a form of system modeling in the concept of Industry 4.0 [60] and digital twin simulations [45]. The importance of this field is also proven by the development of RAMI 4.0 [60] and the other most widely used industrial system models that are characterised like ADACOR (ADAptive holonic COntrol aRchitecture) for distributed manufacturing systems [61].

Summarizing the challenges of ontology modeling and analysis of manufacturing processes are the followings:

- data processing, extraction and interoperability
- contextualization

- standardization in the industry
- modeling of manufacturing processes
- horizontal and vertical integration
- share information
- knowledge representation
- improve the efficiency of assembly processes
- process planning

One of the central questions of the digital transformation is how a production system can be utilized to fulfil all the requirements of Industry 4.0 while following state-of-the-art developments. After introducing general ontologies and semantic networks, the following subsection focuses on applying industry standards related to modeling and data access.

I.1.2 Industry standard-based representation of manufacturing

The application of a standard can improve the enterprise and manufacturing processes of a company in many ways, such as by reducing costs, enhancing how efficient the flow of information between stakeholders and different enterprise levels or human and physical segments, and handling data management challenges [62]. Figure I.1.2 summarizes the relevant international standards available in connection with smart manufacturing, which is categorized according to their fields of application. At the intersection of *Business and Supply Chain Logistics* and *Manufacturing Operations Management* (Figure I.1.2), the ANSI/ISA-95 (IEC 62264 from International Electrotechnical Commission) standard is advised, which uses a five-level hierarchical control model to represent the Business Logistics, Manufacturing Operations Management, Production Control and Production Process functions [63, 64]. It is a widely used international standard produced by the International Society of Automation for developing an automated interface between enterprise and control systems [5]. The ISA-95 can be potentially related to the creation of a manufacturing process ontology or knowledge graph, like:

- Product/Process/Model hierarchy
- Product capability model
- Role-based equipment hierarchy

The different model parts from IEC 62264 are linked together logically to define the hierarchy of sub-models as shown in Figure I.1.3. The production information defines *what was made and used* in the process, that is, which elements correspond to information during production scheduling that listed *what was to be made and used.* The production scheduling elements correspond to the product definition that shows what is specified to make a product. The process segment descriptions are defined by the product definition elements that prove what can be done with the production resources according to the information available. Process Specification and Production Capability prove the main information about the resources [65].

Figure I.1.4 shows a UML (Unified Modeling Language) [66] diagram describing the Production Capability model regarding IEC 62264, the information from sub-classes represent the capability and capability property characteristics of Personnel, Equipment and Materials.

In Figure I.1.5, the Personnel model is represented with a UML diagram, which contains information about the *Class* type of person in the enterprise, such as a production manager or operator; the *Property* as seniority, position, or division; and *Qualification* such as a special task or position of the Personnel.

There are several new methods/frameworks for the standardization and modeling of modern production automation such as IICF (Industrial IoT Connectivity

FIGURE I.1.2: A summary of international standards for Smart Manufacturing [62]

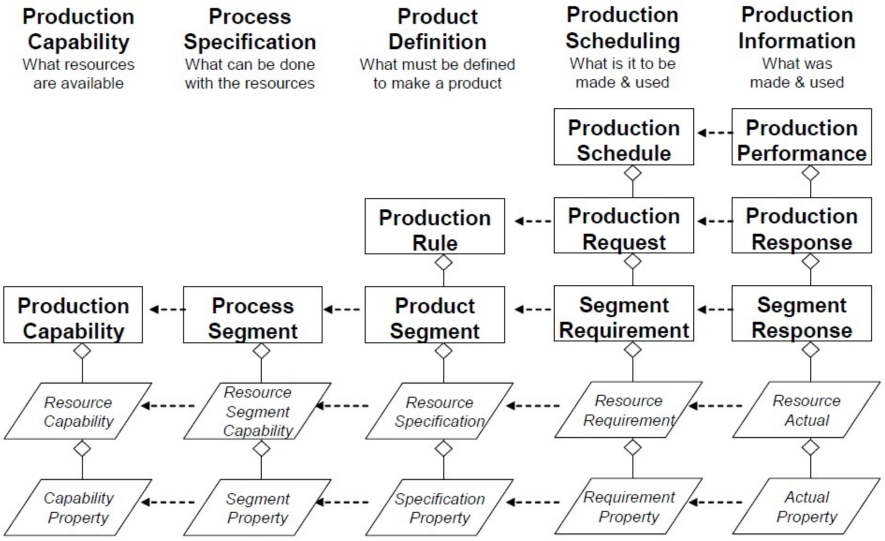


FIGURE I.1.3: The IEC 62264 hierarchy model (IEC 62264) [65]

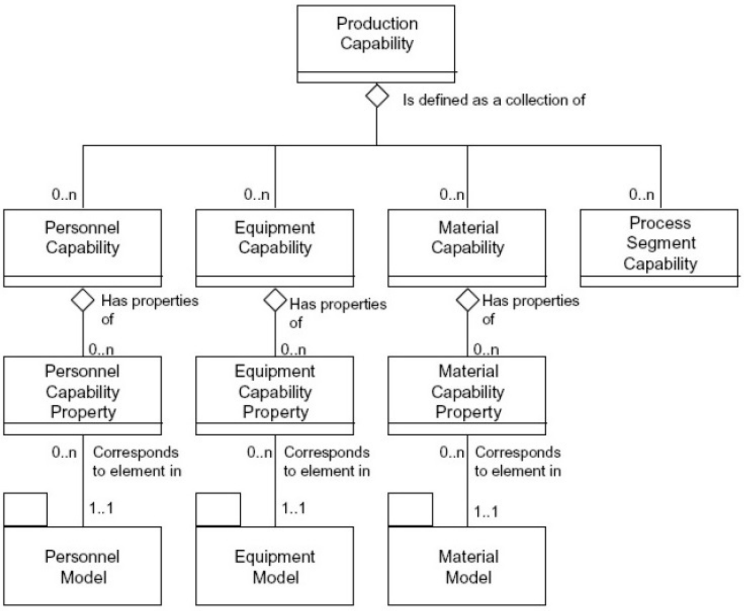


FIGURE I.1.4: The Production Capability Model (IEC 62264) [65]

Framework), IIRA (Industrial Internet Reference Architecture) [68], NIST (National Institute of Standards and Technology) [69], and RAMI [70]. The basic principle of these methods are similar to the ISA-95 standard; it organizes the production/manufacturing processes into a unified hierarchy, allowing for the appropriate communication of information.

Additional standards to consider in connection with semantic technologies and manufacturing modeling are the following:

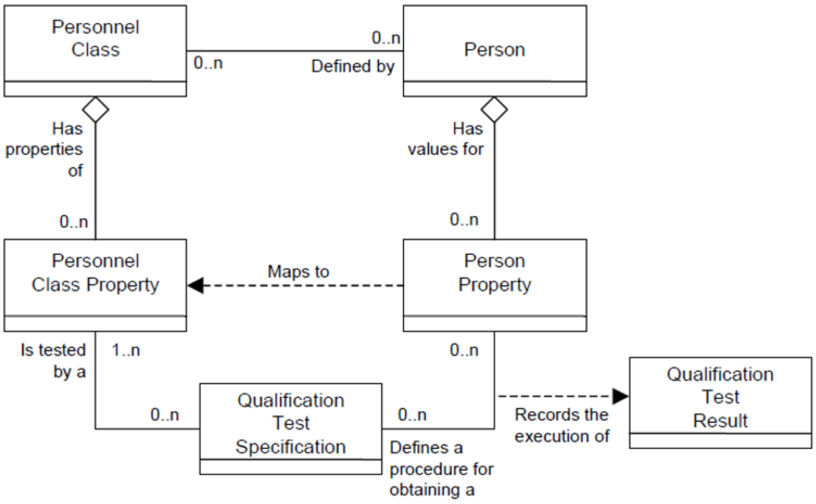


FIGURE I.1.5: Personnel model from ISA-95 [67]

- **ISO 15926** [71, 72] specifies an ontology for asset planning of process plants and an XML (Extensible Markup Language) schema derived from the ontology to exchange the data used for asset planning, including oil and gas production facilities.

- **IEC 62264** [73, 65] defines generic logic models to exchange product and process information between business and manufacturing levels of enterprise applications. Additionally, it enables the manufacturing operations to be integrated with the control domain as an international standard for enterprise-control system integration.

- **ISO 23247** [74, 75] provides an overview and general principles of a digital twin framework for manufacturing systems, including all the terms, definitions and requirements.

- **IEC 61499** [76] defines a generic architecture and guidelines to function blocks in distributed industrial-process measurement and control systems (IPMCSs), moreover, offers textual syntax and graphical representations.

The B2MML is a highly utilized implementation of IEC/ISO 62264 to provide a freely available XML for manufacturing companies [77]. In a standard B2MML modelm the operator is described as *Person* as an XML schema, which is an element of the *PersonnelClass*, and extendable with properties, such as *PersonProperty*, *Location*, *PersonType* or *PersonnelCapability*. Furthermore, a *JobOrder* schema element is also can be interlinked in the model with an operator, where information as *WorkType*, *Priority*, *Command*, *PersonnelRequirement* or *OperationLocation* can be stored. B2MML standard elements are recommended for developing problem-specific ontologies, as the concept of collaborative assembly workplaces [78], where semantic technologies are utilized to enhance interoperability with external legacy systems such as ERP and MES. The so-called *VAR ontology* has three main parts, the tangible assets, the intangible assets and the dynamic status. Another example of adapting the B2MML standard elements in a problem-specific ontology has been published in a paper [78], about a concept of collaborative assembly workplaces, using semantic technologies in order to enhance interoperability with external legacy systems such as ERP and MES.

AutomationML [79] aims to standardize data exchange in the engineering process of production systems. In an AutomationML environment the IEC 62264-2 personnel model [80] offers a method to model the operator in a production process with the following elements: *Personnel Class*, *Personnel Class Property*, *Person* and *Person Property.* AutomationML is also advised for an exchange file format to be a step of automatic workplace design-based on optimized resource allocation [81]. The so-called product-process-resource-triplets (PPR) [82] are a set of appropriate and feasible resources for the assembly steps and the additional product requirements. The creation of the PPR-triplets based on the *workplace*, *products*, *processes* data, which can be stored in AutomationML file format. The mapping information of PPR can assist derive the processes and resources required to manufacture the designed product.

Integrating the industrial aspects with linked data and semantic technologies leads us to the field of knowledge graph solutions. Therefore, the following sub-section discusses semantic modeling in manufacturing.

I.1.3 Semantic modeling, ontologies and description methods of manufacturing systems

In this sub-section, the goal is to provide a simple overview of describing the previously mentioned production standards using the most critical elements of semantics & syntax. This section provides the necessary theoretical background to understand the specific semantic models used in ontologies and description methods of a manufacturing system. Furthermore, the most relevant ontological methods are summarized to describe production systems within the scope of Industry 4.0. The RAMI and the AutomationML framework are discussed more widely, as the literature review shows that leading state-of-the-art research is related to RAMI.

The Semantic Web Stack (see Figure I.1.6) illustrates the hierarchy of (web) languages, where each layer uses the capabilities of the layers below it. It represents how technologies (which are standardized for the Semantic Web) are organized to make the Semantic Web possible [83]. A realistic architecture for the Semantic Web must be based on multiple independent but interoperable stacks of languages [84, 85]. Ontology and knowledge-graph development require these synergistic syntaxes in the same way.

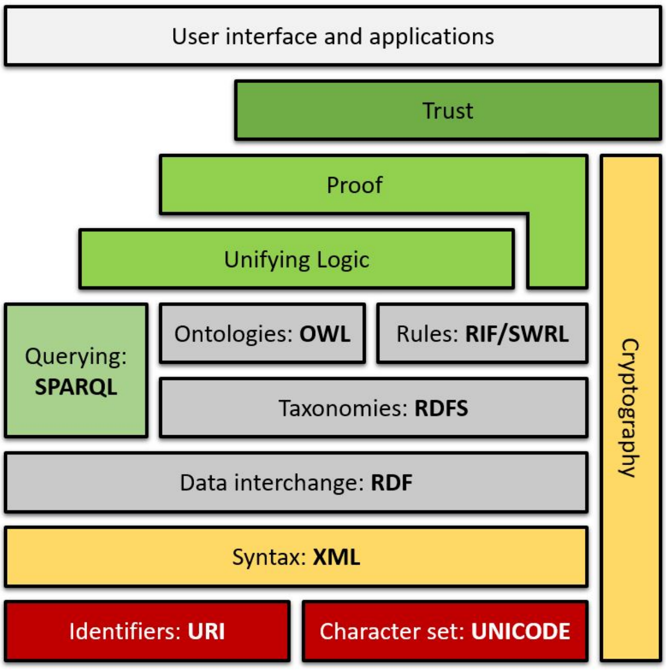


Figure I.1.6: The Semantic Web Stack [83]

The two main building blocks are the RDF (Resource Description Framework) [86] and OWL (Web Ontology Language) [87]. The RDFS (RDF Schema) provides interoperability between applications that exchange machine-understandable information on the Web via standard data representation. It has a wide range of applications, for example, resource discovery to provide better search-engine capabilities or to describe the content and content relationships available on a particular website. OWL can develop domain-specific schemas and ontologies (so-called meta-models) and represent the meaning of terms in vocabularies and the relationships between such terms.

RDF triples can be utilized to extend a graph between unique data instances, collect general data as well as express semantics, attributes and schemas [88]. Complex RDF-based databases are called as triplestores. Another principle uses URIs (Uniform Resource Identifier) to link data by creating triple sentences with subjects, predicates and objects [89]. Figure I.1.7 shows a visualized example. The Operator plays the role of the subject, which has a Data property link (about) to additional data concerning the Operator data, Skill or ID. Furthermore, information about the subject is provided by a predicate that is linked to an object using an Object Property, so the RDF Operator (Subject) assigns an Operator (Object Property) to the Activity (Object).

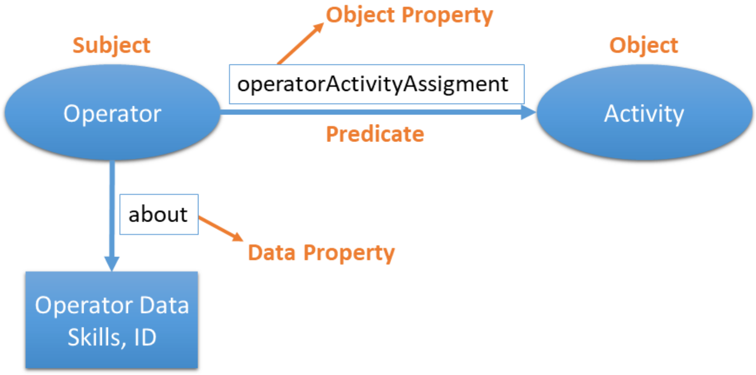


FIGURE I.1.7: Theoretical example of an RDF triple in a RDF model

Nowadays, the most utilized ontology language for Semantic Web applications is *OWL 2* [90], the structure of which is illustrated in Figure I.1.8. The main building blocks of *OWL 2* are various concrete syntaxes that can be used to serialize and exchange ontologies. Each part of the Semantic Web Stack (Figure I.1.6) can be accessed with *OWL 2*. Therefore, the application of this language is regarded as a highly versatile and well-applicable development method for ontologies. An optional connection is highlighted by the Manchester syntax in Figure I.1.8, which stands for the capability of *OWL 2* to write database queries using SPARQL (Structured Protocol and RDF Query Language) to manage knowledge explorations in ontologies [91].

Standards have significant importance for realizing the Industry 4.0 vision and industrial digital chain monitoring to reduce costs. There are several studies [92, 93]

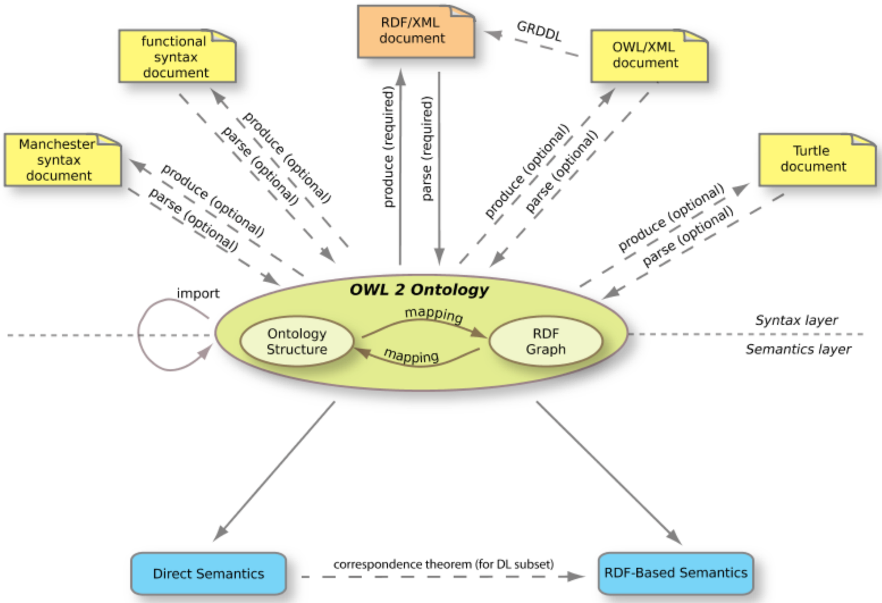


FIGURE I.1.8: The structure of *OWL 2* (Web Ontology Language) [90]

concerning the management of Industry 4.0-related standards and terminologies as well as the creation of knowledge-based frameworks. A good overview is provided by an ontology called *Industry 4.0 Knowledge Graph* [94], which has been developed in order to represent and categorize standards, standardization organizations and standardization frameworks involved in the domain of Industry 4.0 [93]. In Figure I.1.9, the complexity of this field is highlighted. Different domains are connected to the RAMI 4.0 & Asset Administration Shell [95] such as Hierarchy Levels, Communication Layers, Engineering and Semantics.

The Asset Administration Shell has also been established to provide a digital representation of all related information and services involved in manufacturing components [96, 70]. These layers are listed and categorised in Figure I.1.9. In this part of the thesis, the focus is on Semantics and Hierarchy as building blocks of RAMI 4.0 & the Asset Administration Shell, as is highlighted in red in the figure. The Semantic Web Stack of the W3C (World Wide Web Consortium) is a significant building block of the RAMI 4.0 system (see Figure I.1.9).

Furthermore, some examples will be shown to prove the importance of the semantic representation of industrial standards and processes. Therefore, the studied semantic and descriptive methods are summarized in Table I.1.1.

In the modern industry, one of the most widely used frameworks to describe components of CPSs from various perspectives is the AutomationML standard [79].

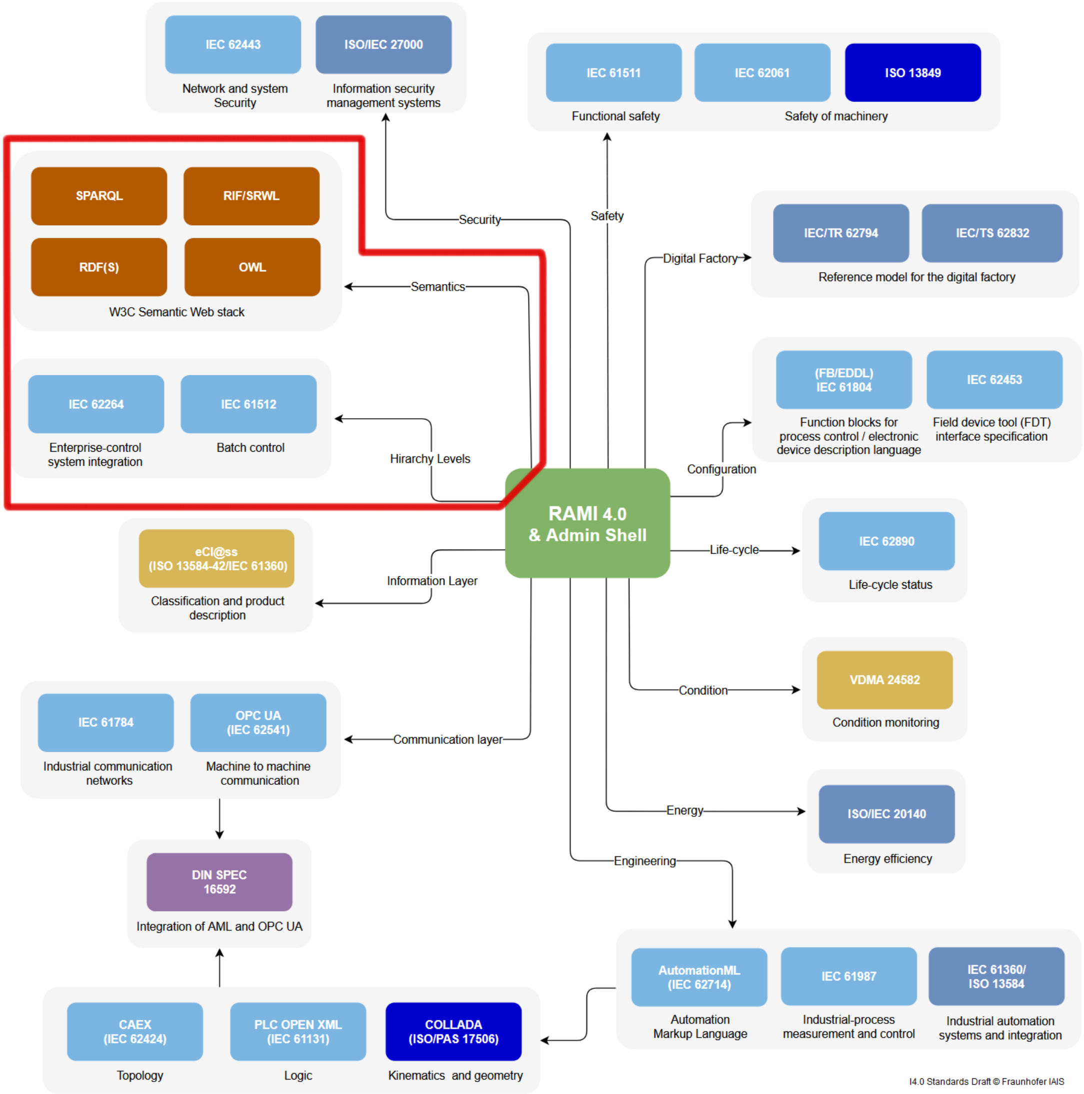


FIGURE I.1.9: Industry 4.0 standards by means of Semantic Technologies [93]

An open, XML-based data exchange format aims to ensure consistent and lossless data exchange in the design of manufacturing systems. It enables systems to be modelled from single automation components to entire large and complex production models. It supports the representation of various aspects of the system, namely topology, geometry, kinematics and control behavior [79]. In addition, AutomationML methods are capable of modeling IEC 62264-2-compliant information too [80]. Furthermore, a specific ontology has also been developed, namely the *AutomationML Ontology* (AMLO), to provide a semantic tool for the improvement of engineering processes in the design of CPSs [97].

The *Sensor, Observation, Sample, and Actuator* (SOSA) ontology provides a core

Table I.1.1: List of the most relevant ontologies and description methods in manufacturing

Name	**Short description**
I40KG [94]	Industry 4.0 Knowledge Graph
AutomationML [79]	A standardized XML-based Automation Markup Language, which aims to store and exchange the information of plant engineering.
AMLO [97]	AutomationML Ontology, which covers the Computer Aided Engineering Exchange (CAEX) section of the standard.
SOSA & SSN [98]	Sensor, Observation, Sample, and Actuator & Semantic Sensor Network ontologies
SWE [98]	Sensor Web Enablement, which is a suite of standards that has been developed
MaRCO [99]	Manufacturing Resource Capability Ontology
IoT-O [100]	Internet of Things Ontology
BFO [101]	Basic Formal Ontology
ASP [102]	Assembly Sequence Planning ontology
MSDL [103]	Manufacturing Service Description Language
DOLCE [104]	Descriptive Ontology for Linguistic and Cognitive Engineering
DUL [105]	DOLCE-UltraLite, which is an upper and extended ontology of DOLCE

for *Semantic Sensor Network ontology* (SSN) as well as extends the target audience and application areas, making use of Semantic Web ontologies. It has been used as part of an architecture for the Web of Things, sensing in manufacturing, representing humans and personal devices as sensors, as well as part of a linked data infrastructure for SWE (Sensor Web Enablement) [98].

The *Manufacturing Resource Capability Ontology* (MaRCO) supports the rapid semi-automatic system design, reconfiguration and auto-configuration of production systems. MaRCO has been developed for the quick identification of candidate resources, and resource combinations for a specific production need [99]. *IoT-O* is a core-domain modular IoT (Internet of Things) ontology that proposes a vocabulary to describe connected devices and their relationship with their environment. It describes concepts like Electronic Device, Smart Network, Smart Entity, Physical Entity, Control Entity, and so on [106, 100].

Upper ontologies can define top-level concepts such as activities, physical objects or topological relations from which more specific classes and relations can be defined. Engineers, which Upper ontologies are used to develop a more specific domain ontology by starting with the identification of crucial concepts utilizing activity modelling, use cases, and competency questions [72]. The following discusses the more relevant upper ontologies concerning manufacturing systems.

The *Basic Formal Ontology* (BFO) serves as an upper-level framework, which has been developed to assist the organization and the integration of data obtained through scientific research [101, 107]. Furthermore, BFO is currently undergoing a certification process with the International Organization for Standardization (ISO) as a top-level ontology for information technology [108].

Assembly Sequence Planning (ASP) ontology formally defines the assembly knowledge, where all the assembly knowledge in the sequence generation approach is expressed and stored. ASP determines the sequences and paths of parts to assemble a product with minimum costs and over the shortest period of time [102, 109].

Manufacturing Service Description Language (MSDL) provides the simple building blocks required to describe a broad spectrum of manufacturing services. MSDL is a description of the manufacturing capabilities of manufacturing resources at different abstraction levels, namely machine, workstation, cell, shop and factory [103].

A widely used ontology with many extensions is the *Descriptive Ontology for Linguistic and Cognitive Engineering* (DOLCE). The DOLCE upper ontology aims to capture the ontological categories underlying natural language and human common sense, which is of great importance in terms of the Semantic Web [104]. *DOLCE+DnS Ultra Lite* (DUL) is the OWL version of DOLCE, extended to cover the Descriptions and Situations (DnS) framework, and is a widely adopted ontology in projects worldwide. The foundational concepts of DOLCE can be utilized for aligning domain ontologies, e.g. for Semantic Sensor Networks [110]. DUL is also used for creating a formal model of events to provide comprehensive support to represent time and space, objects and people, as well as causal or correlational relationships between events [105]. With another DUL application, the extraction and description of emerging content ontology design patterns are achieved [111].

In summary, ontologies play a vital role in developing smart factory concepts [112], and the application of their elements as namespaces or vocabularies in specific manufacturing-related ontologies is justifiable.

I.1.4 Product-process-resource modeling and workflow

This subsection gives a brief overview of the PPR (Product-Process-Resource) modeling and workflow concept as well as their utilization.

Product-Process-Resource (PPR)-based modeling is compatible with AutomationML and serves as an approach for creating a knowledge-driven product, process and resource mappings in assembly automation [82]. The aim is to reduce the development time and engineering costs by enabling collaboration within sectors as well as realization of product and manufacturing resources in a virtual environment. The main benefit of PPR-based modeling is to manage the mapping of engineering data sets as well as interconnect product attributes with manufacturing processes and resources. Additionally, knowledge-based PPR mapping can be utilized for dynamic configuration and the analysis of assembly automation systems [113].

In the PPR-triplets format, the information is described by three different domain ontologies using the CPS knowledge repositories, namely Product Ontology, Process Ontology and Resource Ontology [114]. Additionally, an approach has been developed to manage modularity in production management based on the PPR ontology [115] and the ISO 15531 MANDATE standard presented for exchanging industrial manufacturing management data. The PPR approach also has been extended with definitions of skills, creating the PPRS (product, process, resource, skill) [116] model. The aim of PPRS is to exchange information about skill-based adaptive production systems.

Another implementation of PPR modeling is the description of workflows. A PPR-based model for an engineering workflow that aims to support factory automation is presented in Figure I.1.10 [117]. The model defines four layers, that is, Team, Tool, Middleware and Knowledge Base, as well as three domains, namely Product, Process and Resource. Furthermore, the key tasks of such an engineering workflow are labeled on the right-hand side of the Figure.

As the market must be continuously re-engineered and modern manufacturing systems reconfigured, ontology modules that create functional links in the engineering workflow are necessary. This architecture combines digital engineering tools, standards, data models and a modular knowledge base to link the activities

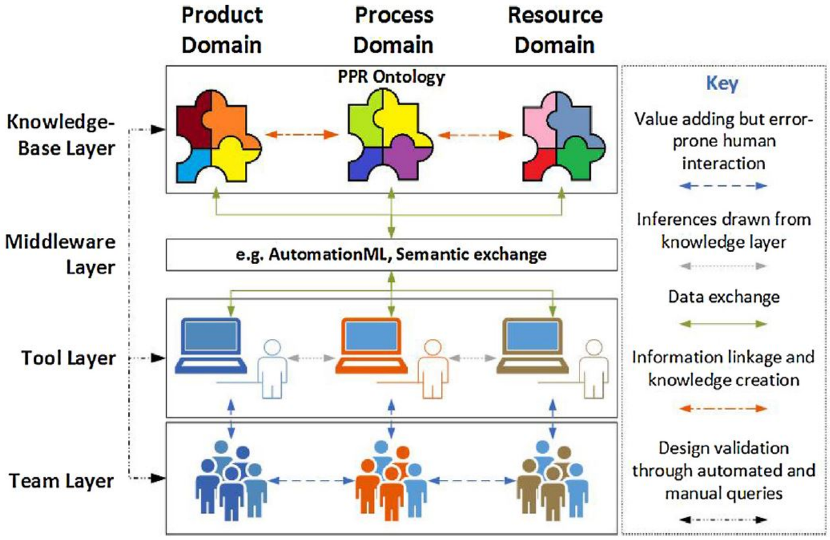


FIGURE I.1.10: Product-process-resource modeling-based conceptual engineering workflow [117]

as well as information across the PPR domains, as is shown in Figure I.1.10. Additionally, AutomationML [79] is highlighted as an ideal intermediate layer to the approach.

Regarding ontology modules, the OWL model of representing manufacturing data allows the semantic description of each component to be inherited, extended or adapted. Therefore, in the following subsection, some specific ontological, semantic-based and KG solutions to support operators are discussed, starting with human activity recognition.

I.1.5 Semantic technologies and metrics to describe and support the operator

This section summarizes the relevant semantic technologies and metrics related to the Industry 5.0 and operator support. First, subsection I.1.5.1 presents the topic of human activity recognition. Subsection I.1.5.2 gives some examples of ontology-based modeling and support systems for ergonomics and collaboration. In subsection I.1.5.3, the field of quality metrics to evaluate human-machine interactions is discussed.

I.1.5.1 Human activity recognition

This subsection proposes some recommendations and examples of applications for HAR solutions to support operators.

The design challenges of a HAR system proposed by a survey [118] are the following: (1) selection of attributes and sensors, (2) obtrusiveness, (3) data collection protocol, (4) performance recognition, (5) energy consumption, (6) processing and (7) flexibility. During the development of a human-centered KG, each of these aspects has to be considered. In smart factories, wearable sensors are one of the most significant emerging technologies, which can be highly utilized to support operators and perform activity recognition. Regarding the nature of sensors, whether wearable or external, a HAR system can be online, supervised offline or semi-supervised [118]. Such devices, for example, can be indoor positioning systems, heart monitors or light sensors.

Different methods and ontologies for human behavior recognition can be classified as data-driven and knowledge-based techniques. The integration of these two methodologies is recommended to help manage limitations in scenarios with several actors, provide semantics in a variety of production activities or for the purpose of worker identification according to behavioral semantics [119].

Utilization of a machine learning-aided approach has been proposed, where online activity recognition and activity discovery are combined in an algorithm [120], moreover, the method identifies patterns in sensor data, which can provide insights into behavioral patterns. The approach can be used to identify and correct possible sources of annotation errors, thereby improving the quality of the annotated data.

Another recent paper shows how a machine learning semantic layer can complement augmented reality solutions in the industry by providing a so-called intelligent layer [121]. The method can validate the performed actions of the operators, such as checking whether the operator has activated a specific switch before moving on to the next step. Additionally, operator assistance is possible with this semantic solution, e.g. to allow the operator to access valuable context information in natural language [121].

In semantic-based human activity recognition, one of the most significant features is its ability to recognise new activities that have not been pre-stored or trained previously in the system. In a paper in which activities were recognised from image and video data with semantic features, combined with deep learning image analysis [122], the activities were divided into four groups, namely atomic actions, interactions between people, human–object interactions and group activities. Furthermore, the most popular features of an action should be included in the semantic space, e.g. the human body and poses, attributes, related objects or context of the scene.

Concerning activity recognition, it is also important to mention situation awareness, which is a critical modeling element. Situation awareness heavily relies on the knowledge of relations. The advantage of an ontology-based approach with regard to situation awareness is that once facts about the world have been stated in terms of the semantic network, other facts can be inferred using an inference engine. The Situation Theory Ontology (STO) has been developed to express the situation theory as a formal OWL ontology and provide computer-processable semantics in situation theory [123]. Additionally, a study investigated the analysis of eye movements to evaluate situation awareness in human-robot interactions [124].

After investigating the human activity recognition solutions, in the following subsection, the support of the operator is discussed from the view of ergonomics and human-machine collaboration.

I.1.5.2 Ergonomics and collaboration

This subsection highlights the importance and benefits of integrating ontologies into a human-centered knowledge graph, which facilitates ergonomics and collaboration in a production environment.

Ontology evolution must be supported through the entire life cycle. The *Human-Centered Ontology Engineering Methodology* [125], following the human-centered approach, strongly highlights the integration of ontology engineering environments with the practices of knowledge workers, enabling knowledge workers to interact directly with their conceptualisations at a high level of abstraction.

The operators must be allowed to easily interact with industrial assets while working on other more complex ones in an Industry 5.0 environment. To fulfill this development goal, a generic semantics-based task-oriented dialogue system framework such as KIDE4I (Knowledge-drIven Dialogue framEwork for Industry) [126] may offer a solution to reduce the cognitive demand. The more process steps that can be made easier in terms of production with voice or motion control, the more the procedures can be simplified for the operator and the more ergonomic a work environment can be. Additionally, the takt times can be shortened thanks to the developed features concerning human-machine interaction.

The ergonomics system can be divided into three subsystems, that is, the human, machine and environment as well as the monitored elements and conditions of them [127]. The physical load stands for how much manual labor the operator is able to handle without decreasing their work efficiency, while mental load describes the psychological pressure and information processing while working. In the design of modern production space, it is essential to monitor several factors in the environment as well as on the machines and devices, as in the aforementioned example, to observe as many physical and mental characteristics of the personnel as possible. By embedding these parameters into the KG, efficient human-machine collaboration and more ergonomic workspace could result with continuous improvement.

As evidenced in a previous study [128], it is recommended that a multi-ontology approach and the Cynefin Framework [129] be applied with regard to ergonomics, multiple views and interaction between multiple agents. In this approach, four domains are used, namely the simple, the complex, the complicated and the chaotic, to provide a way of re-perceiving situations where ergonomics-related problems can occur or have already been identified. The design of an ergonomic work environment with a multi-ontology methodology could also facilitate human-machine collaboration.

An additional aspect to mention is the integration of cyber, physical and socio-spaces through Industry 4.0, leading to the emergence of a new type of production system known as cyber-physical production systems (CPPSs). A paper that studied human-centered CPPSs in smart factories and active human-machine cooperation proposed an ontological framework, the PSP Ontology (Problem, Solution, Problem-Solver Ontology) [130]. The investigated problem linked the three super-concepts of “Problem-Solving Semantically Profile”, “Problem-Solver Profile” and “Solution Profile”. Besides the semantic representation and reasoning of the super-concepts, they proposed the contingency vector, the vectors of competence and autonomy as well as the solution maturity index for CPPS [130].

After discussing the human-centric ergonomics and collaboration features, the following subsection presents the relevant metrics to evaluate these factors and processes.

I.1.5.3 Metrics to evaluate human-machine interactions

In an Industry 4.0 environment, tools and techniques for monitoring as well as supervising the performance of CPS and H-CPS systems are essential. Metrics, e.g. Key Performance Indicators (KPIs) and Human-Robot Interaction (HRI) factors, are a set of parameters that permit the evaluation of the performance of a specific asset, system or worker. Therefore, this subsection briefly describes the importance of this field as well as presents some examples and recent applications in smart factories.

The KPIs related to MOM (which were previously discussed in Section I.3.1) are part of the ISO 22400 standard as "Key performance indicators for manufacturing operations management" [131]. A study [132] presented a method for implementing and visualizing these ISO 22400-based KPIs, which are described by an ontology. The description is done according to the data models included in the KPI Markup Language (KPIML), which is an extension of AutomationML as an XML implementation developed by the international organization Manufacturing Enterprise Solutions Association (MESA). The ontology-based KPI framework and visualization features are presented in a study [132] that consist of five elements: Knowledge-Based System Service, Manufacturing Plant, Orchestration Engine, KPI Implementation, and User Interface. Additionally, the approach can be utilized to visualize the KPI as well as provide event notifications at a manufacturing plant and updates of knowledge from and to an ontology for the users.

An important additional question to discuss regarding KPIs for manufacturing operations management is whether the defined KPIs are perfectly suitable for the process industry in the ISO 22400 standard. A gap analysis [133] within the ISO 22400 standard and the industrial needs of the process characterised the gaps into three main categories, namely (1) only a few of the defined KPIs are suitable for the process industry, (2) the relationships as well as working conditions of each unit are different, and (3) some defined KPIs cannot be computed or are even meaningless. Moreover, it has been concluded by the analysis that the indicators

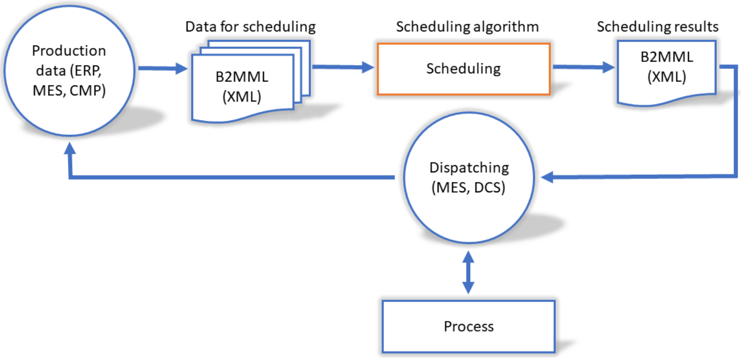


FIGURE I.1.11: Flexible scheduling supported by production ontology data [134]

appear to be primarily designed for discrete industries and ISO 22400 cannot meet the requirements of the practical production process.

After evaluating the performance and interactions between factors, the following subsection investigates the possible applications of semantic technologies for the purposes of optimization and decision-making support.

I.1.6 Ontology-based analysis and solutions in manufacturing systems

In this section, it is shown, through a couple of applications, that ontology- and knowledge graph-based solutions for production are no longer just concepts but methods that have already been implemented in the industry.

A research group of ABB company proposed a study about a scheduling solution connected to a production environment aligned with the ISA-95 standard and using B2MML to share information. The methodology of the ontology data-supported workflow is shown in Figure I.1.11 [134]. The development of Enterprise Control Ontology (ECO) [135] also exemplifies well that semantic models provide various solutions to address operational issues in production. In ECO, more sub-ontologies have been combined to create a manufacturing systems model, which provides domain information about entities and enables the reconfiguration of manufacturing systems.

Another application example is the *SemCPS framework* (Semantically Described Cyber-Physical Systems) for enabling the integration of CPS descriptions in knowledge graphs. The approach can effectively integrate CPS perspectives using Uncertain Knowledge Graphs of smart manufacturing-related standards such as AutomationML [136]. Numerous application and research examples can be found for AutomationML-related developments, like data modeling and Digital Twin Exchange [137, 45], or IEC 62264 standard-based AutomationML models can be created [138]. Another important aspect is to implement the Bill of Process (BOP), and Bill of Materials (BOM) in data models to map resources or abilities in order to perform a task based on physical skills [139]. The integration of BOP and BOM in resource description models can support assembly planning engineers and provide a framework for clustering production information [140].

The Uniform Project Ontology utilizes linked data and the Semantic Web in order to represent knowledge about mega projects to facilitate data processing and utilization through their entire life cycle [89]. The purpose of a recently published modular domain ontology is to describe the cyber and physical aspects of automation systems that support simultaneous engineering [48]. The effectiveness of the knowledge-based approach to designing assemblies in agile manufacturing to integrate a linked product and process data has been studied and proven [141].

A study by Siemens has proven that semantic technologies can improve the feature selection method for machine learning models in industrial automation systems in order to reduce the size of feature spaces for data labeling problems using only a small amount of semantic relations [142]. During the Optique program [143], the goal was to develop an Ontology-Based Data Access (OBDA) system and provide access to Industrial Big Data stores using Semantic Web technologies. An impressive demonstration of using Optique's OBDA system customised for the user is Siemens Energy's data access challenge. The technology solution was used to answer questions concerning data queries as follows "Return the TOP 10 errors and warnings for turbines of product family X" or “Which events frequently occur before a specific point in time?” [144]. A further project in collaboration with Siemens studied the application of ontologies to create industrial information models in manufacturing and energy production. It led to the development of the Siemens-Oxford Model Manager (SOMM) tool to support engineers in creating ontology-based models [145].

The applicability of a graph-based framework for advanced manufacturing analytics has been also demonstrated by representing manufacturing data as a multigraph using a semantic abstraction layer to integrate flexible data. Herewith provides a tool for predictive and prescriptive decision making by detecting fault patterns [146].

The methodology of RDF triples (discussed in Section I.1.3) how links data is the same as in the case of bipartite graphs. Studies are published to formalise the bipartite graphs as an intermediate model for RDF with a goal of graph-based notions in querying and storage [147]. Furthermore, an RDF database can be simultaneously analysed as layers of a multilayer network [148], providing a solution for Production Flow Analysis.

One of the main challenges of manufacturing optimization is intelligent resource allocation, which aims to transform data into knowledge while optimizing the allocation between personalised orders and manufacturing resources. A study [149] investigated how the complex data of workshop resources can be fully integrated and the implicit semantic information mined to form a viable knowledge-driven resource allocation optimization method. The Workshop Resource Knowledge Graph (WRKG) model has been developed to integrate the semantic engineering information in the machining workshop. Moreover, a novel knowledge graph-based resource allocation optimization approach for a device has been proposed to mine the implicit resource information for updating the WRKG in real-time [149]. Therefore, the research presented a unified knowledge graph-driven production resource allocation approach, allowing fast decision-making regarding resource allocation for given tasks concerning the insertion of orders in light of the resource machining information and the device evaluation strategy [149].

Another study focused on the closed-loop optimization possibilities of a knowledge graph [13]. Better data exchange and representation are demanded, as the application of different data transfer protocols results in scalability issues (when integrating new hardware) as well as software and interoperability issues (when collaborating between different platforms). A dynamic knowledge graph-based approach towards automated closed-loop optimization has been proposed [13], which consists of three layers, namely the real world layer, dynamic knowledge graph layer and active agents layer. The middle layer is dynamic as it reflects and influences the status of the real world in real-time. The study of closed-loop optimization and knowledge-graph development concluded that a dynamic knowledge graph-based approach would enable rapid integration of data and AI-based agents for the purposes of data discovery as well as development [13].

The knowledge graphs and graph embedding also can be utilized to develop recommendation techniques [150]. Such a method can offer an information representation technique alloying content-based and collaborative information. Furthermore, the recommendation systems can facilitate the writing of SPARQL queries in a semantic system, which is beneficial as the scheme of the dataset is usually not known in advance [151].

An ontology module-based framework has been developed to support the self-evolvability and self-configuration of production systems. The method provides the possibility to represent the required information in module modeling based on generic information standards together with the ontology of the modules, using the semantic Reference Data Library (RDL) [152].

A recent paper proposes an ontology-based decision support system to assist the reconfiguration of manufacturing systems with multi-criteria decision making [153], which combines semantic technologies with Technique for Order of Preferences by Similarity to Ideal Solution (TOPSIS). Additionally, as the ontology can describe production requirements, disturbances and configurations, it offers a basis for a Case-Based Reasoning (CBR) approach for system reconfiguration. The knowledge-based reconfiguration of manufacturing systems is based on expert knowledge captured by an ontology, which is used both to monitor the manufacturing system and recommend configurations [153]. Additionally, the system is compatible with industrial enterprise information systems such as ERP or MES as well as with legacy decision support software such as quality control and maintenance tools.

An additional aspect of the semantic-based application is ontology-based simulation, which aims to fully automate digital twins. A recent paper published an open-source, fully ontology-based discrete-event simulation [154], which also has a user interface to analyse the agents in detail using KPIs. Another publication investigated the transformation of semantic knowledge into a simulation-based decision support system [155].

A knowledge reasoning framework has been proposed, that utilizes semantic data to improve real-time data processing in a smart factory setting [156]. The framework uses an ontology-based knowledge representation method and a rule-based reasoning engine to enable intelligent decision-making and optimization of factory operations. To handle the real-time nature of the data, a stream processing engine has been employed, that processes data in small batches, enabling real-time data analysis and decision-making [156].

A hybrid semantic annotation, extraction, and reasoning framework for Cyber-Physical Systems has been proposed, that aims to enable the development of a Semantic Web of Things (SWoT). The framework utilizes both ontology-based semantic annotation and natural language processing techniques to extract and reason data in a CPS environment, which can improve the interoperability, automation, and decision-making capabilities of these systems [157].

A semantically-enhanced rule-based diagnostics language called SDRL for the Industrial Internet of Things (IIoT) has been presented [158], which leverages Semantic Web technologies to improve the accuracy and efficiency of diagnostics in industrial systems. The authors also present a case study involving Siemens trains and turbines to demonstrate the effectiveness of the proposed approach, which resulted in improved diagnostic performance and reduced maintenance costs [158].

Based on the above-listed application examples, the benefits of the ontology-based features are the following:

- support flexible scheduling and solve operational issues in manufacturing, such as resource allocation [134, 135]
- integrate CPS and describe cyber and physical parts of automation systems [136, 157]
- model Digital Twins and provide access to Industrial Big Data stores [137, 45]
- conceptual design of a data- model and warehouse [138, 139, 140]
- facilitate project lifecycle analytics of mega projects [89]
- effectively support assembly design and planning processes to evaluate risks and costs before the realization of the production processes [48, 141]
- improve the efficiency of machine learning models [142]

- data mining and root cause analysis [143, 144]
- support predictive and prescriptive decision-making [146, 145]
- perform intelligent resource allocation [149]
- closed-loop optimization with integrated knowledge graph [13]
- ontology-based decision support system [153]
- perform manufacturing simulation, using ontology-based discrete-event simulation [154]
- intelligent decision-making with knowledge reasoning on real-time data [156]
- semantically-enhanced rule-based diagnostic for IIoT [158]

After evaluating the advantages of semantic technologies, the following section focuses on the differences with rational databases and also on the limitations of implementing ontologies in industry.

I.1.7 Comparison of ontology-based methods with relational databases and the difficulties of industrial adaptation

This section provides a brief comparison of ontology, and graph-based data processing, with traditional relational databases. Additionally, the difficulties and limitations of utilizing ontology-based methods in the industry is investigated.

A critical review has been presented [159], which studies lifecycle engineering models in manufacturing and compares the use of ontologies and semantic technologies and databases such as RDBMS (Relational Database Management System) [160]. Various ontologies and databases have been analyzed, used in manufacturing and also their strengths and weaknesses have been evaluated. Based on the results of [159], [161] and [162], the following comparison can be made:

- Advantages of ontologies:
  - Semantic interoperability: Ontologies provide a common vocabulary and shared understanding of data, enabling easier integration of data from different sources and applications.
  - Flexible data modeling: Ontologies are more flexible than RDBMS in terms of data modeling, as they allow for the representation of complex relationships and concepts that may be difficult to represent in a tabular format.
  - Knowledge representation: Ontologies can represent not only data, but also knowledge and concepts, allowing for more advanced reasoning and decision-making capabilities.
  - Performance of data queries: SPARQL queries can be more efficient when working with highly interconnected data, as they can take advantage of the graph structure to optimize queries. In contrast, relational databases may suffer from performance issues when working with highly interconnected data.
- Disadvantages of ontologies:
  - Complexity: Developing and maintaining ontologies can be complex and time-consuming, requiring significant expertise and effort.

  - Performance: Querying large ontologies can be computationally expensive, and may not be as efficient as querying a well-designed RDBMS.
  - Data storage: Storing large amounts of data in an ontology can be challenging, as they are typically stored as triples and may not be as efficient as a well-designed RDBMS.

- Advantages of RDBMS:
  - Efficient data storage: RDBMS are well-suited for storing large amounts of structured data, and can be highly optimized for performance and scalability.
  - Well-established technology: RDBMS are a well-established technology with a large user base and extensive tooling available.
  - Familiarity: RDBMS are a familiar technology to many developers and can be easier to work with than ontologies.
- Disadvantages of RDBMS:
  - Data integration: RDBMS can be difficult to integrate with other systems, as they typically rely on a fixed schema and may not be as flexible as an ontology-based approach.
  - Limited semantics: RDBMS are limited in their ability to represent knowledge and concepts, and may not be as well-suited for advanced reasoning and decision-making.
  - Data silos: RDBMS can lead to data silos, where data is stored in different systems that cannot easily be integrated.

An additional important method is the so-called SPARQL-to-SQL translation technique. It is important as in practice most existing RDF stores for large semantic networks, which serve as metadata repositories on the Semantic Web, use an RDBMS as a backend to manage RDF data [163, 164]. Thanks to this technique users can query RDF data stored in a relational database using SPARQL, without having to migrate the data into a triplestore or another RDF database.

Based on the literature review of this Chapter, the following list aims to summarize the possible reasons why knowledge graphs may not be as widely used in the industrial practice compared to other well applied fields as biology [165] or IT [166]:

- Differences in data complexity and diversity: The manufacturing industry deals with a vast amount of complex data, but the data is often structured and transactional in nature, making it easier to manage with traditional databases. In contrast, the field of biology deals with a wide variety of unstructured and semi-structured data, such as scientific papers, clinical trials, and experimental results, which are better suited to be represented using a knowledge graph.
- Focus on operational efficiency: The primary focus of industrial companies is often on optimizing operational efficiency and reducing costs. Knowledge graphs may not be seen as a priority as they are typically used to support strategic decision-making and discovery rather than day-to-day operations.
- Lack of awareness and expertise: The concept of knowledge graphs is relatively new and may not be widely understood in the manufacturing industry. Additionally, the implementation of knowledge graphs requires a certain level of technical expertise, which may not be readily available within the industry.
- Cost of implementation: The cost of implementing a knowledge graph can be significant, particularly for smaller or mid-sized organizations. This may make it more difficult for these organizations to justify the investment.
- Lack of skilled personnel: Building and maintaining a knowledge graph requires specialized skills and expertise, particularly in graph theory, ontology modeling, and RDF data structures. There may be a shortage of personnel with these skills in the industrial sector.

Here are some of the key resources that may be required for constructing, maintaining, and updating knowledge graphs in a production company:

- Data acquisition and integration: Depending on the size and complexity of the organization, the data to be integrated into the knowledge graph may come from multiple sources and formats. As a result, data acquisition and integration can require significant resources, including personnel, hardware, and software.
- Graph design and implementation: The design and implementation of a knowledge graph can require significant expertise in graph theory, data modeling, and database design. This may require hiring specialized personnel or outsourcing the work to a third-party vendor.

- Graph maintenance and updates: Maintaining a knowledge graph requires ongoing monitoring and updates to ensure that the graph remains accurate and up-to-date. This may require a dedicated team of personnel, as well as the use of automated tools and processes to ensure efficiency.
- Hardware and software infrastructure: Depending on the size and complexity of the knowledge graph, hardware and software infrastructure may need to be updated or expanded to support the graph. This may include hardware upgrades, cloud-based solutions, and specialized software tools.

As the benefits of knowledge graphs become more widely recognized and the technology becomes more accessible, it is possible that we may see increased adoption in the manufacturing industry in the future.

In the previous sub-sections, a systematic overview of the ontology-based modeling of production systems has been proposed and summarized; furthermore, collected the most relevant application cases. The following section demonstrates the applicability of the proposed methodology in a wire harness assembly case study.

Chapter I.2

Ontology-based modeling of a wire harness manufacturing processes

This chapter describes an industry related example of semantic modeling and data query analysis, starting with the description of the applied software elements in Section I.2.1. The detailed industrial benchmark is presented in Section C.1 of the Appendix, which is based on a wire harness assembly case study. The ontology-based modeling and the development of a knowledge graph are described in Section I.2.2. Section I.2.3 discusses the data queries with SPARQL and evaluates the query results. Finally, in Section I.2.4, the results and final contributions are presented in detail.

This chapter investigates wire harness manufacturing, which is still highly manual due to the extremely complex maneuvers of the activities [167]. The operators perform various activities with many different workpieces to assemble a complex product. Typically complex modular production systems are applied, where the challenge is that numerous activities and highly manual assembly necessarily require optimum assembly line balancing. As the utilized BOM is also complex, it can serve as a good benchmark problem for ontology-based modeling and analysis.

I.2.1 Applied software tools of ontology-based modeling

Firstly, the different software features are presented, which are utilized in the chapter. The main aspects of software tool selection were to have open-source access and applicability possibility in both research and industry field. The applied software packages involved in this work are as follows:

- Protégé for ontology development and to create the OWL/RDF format [168]
- GraphDB to manage SPARQL queries [169]
- OntoGraph and visualization plugins of the Protégé environment to visualize ontology structure

In the case of data collection of a wire harness manufacturing, a modular assembly system has been studied. The relevant data are the physical parts of the final product, stored in the BOM, the operators at the assembly line and their skill and the necessary equipment. Quantitative factors are the activity times of a particular step of the wire harness assembly and the costs of using different skills or resource-related tools.

For the development of the ontology, the Protégé editor has been used, which is an open-source tool developed by Stanford University to create and edit any ontology [170]. This platform supports all kinds of semantics and data standards like the XML, RDF, or OWL types of ontology datasets [171]. An in-depth knowledge of the manufacturing system behaviour is required to assign which factors or identities will be a Class, Object Property, or Data Property in the formed ontology. For this reason, the hierarchy of manufacturing processes must be adequately reflected using the tools provided by the ISA-95 standard or AutomationML framework (as presented in Section I.1.3). Another critical part of ontology engineering is to find the reusable Ontology and Vocabulary elements, which can be applied in the current ontology. To accomplish this, firstly, industry-specific research papers and semantic solutions (studied in Section I.1.3) can provide a guideline, but also the use of the tool Linked Open Vocabularies (LOV) is recommended as it provides an effective search engine to find adaptable namespaces or vocabulary elements [172].

For data analysis and SPARQL queries, the GraphDB software has been used, which is an RDF-capable database tool, especially for Knowledge Graphs. The most significant benefit of SPARQL queries is creating a structured version from the data stored in the ontology or extracting data from RDF, which is an excellent source to manage basic production analysis. However, if a more in-depth investigation is needed or the production ontology or it has a high complexity, the tools of Data Science can provide more accurate solutions.

Additionally, visualization tools play an essential part in the methodology in several phases of the process. OWL visualization tools has been utilized, which are part of the Protégé [173] as the VOWL (Visual Notation for OWL Ontologies) [174]. The following sections prove the efficiency of the ontology-based modeling methodology with a manufacturing-specific benchmark.

I.2.2 Ontology modeling - Creation of manufacturing based knowledge graph

This section presents the modeling part of the ontology development, based on the theoretical background presented in Section I.1.3. The classes and their interactions are determined regarding the use case of wire harness manufacturing, which is discussed in the appendix, in Section C.1.

As the first phase of the ontology development, the basic structure has to be established. Therefore, different classes are defined as the elements of this specific production process, as shown in Figure I.2.1. To characterise the relationships between these classes, different types of connections are distinguished as cost, optimizable, or technical parameters (highlighted with orange, green and purple color). These properties related to interconnections, so-called Object Properties, significantly determine the characteristics of the production process.

The presented wire harness manufacturing ontology consists of 9 Classes: Product, Module, Component, Activity, Skill, Operator, Workstation, Equipment and Resource (namely electricity for some tools). Furthermore, interactions between these Classes are denoted by arrows in Figure I.2.1, pointing to the Domain Class of the Object Property.

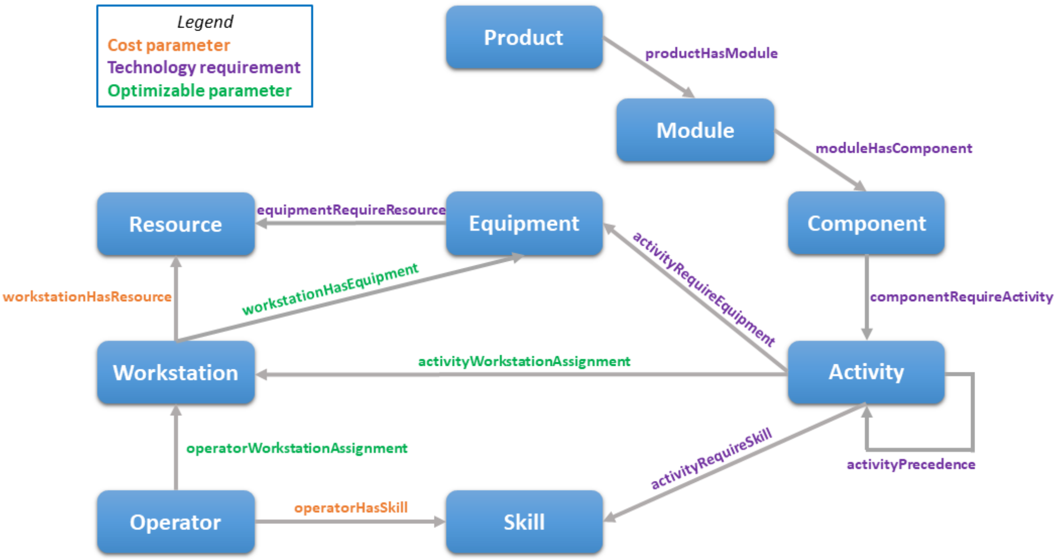


FIGURE I.2.1: Technology and cost-related factors of a wire harness manufacturing visualized on the theoretical ontology

Once the theoretical structure of the ontology is available, the following part is the creation of a manufacturing-based knowledge graph. Where first, the relevant vocabulary and namespaces elements are implemented. Table I.2.1 summarizes the namespaces used for the wire harness manufacturing ontology. The sources of the vocabularies are cited next to their prefixes in the table. Figure I.2.2 represents the structure of the wire harness manufacturing ontology after the integration of vocabulary elements. As listed in the legend, the different Classes and Object

TABLE I.2.1: List of the integrated ontology namespaces.

Prefix	**Vocabulary namespace - Description**
RAMI [175]	RAMI is a Vocabulary to represent the Reference Architectural Model for Industry 4.0.
SMO [176]	Semantic Manufacturing Ontology
PROV [177]	The PROV Ontology (PROV-O) provides a set of classes, properties and restrictions that can be used to represent and interchange provenance information or data coming from different systems and in different contexts.
SCOR [178]	The vocabulary SCORVoc formalises the latest SCOR (Supply-Chain Operations Reference) standards while overcoming the identified limitations of existing formalisations.
DUL [179]	DOLCE+DnS UltraLite ontology aims to provide a set of upper-level concepts that can form the basis for easier interoperability among middle and lower level ontologies.

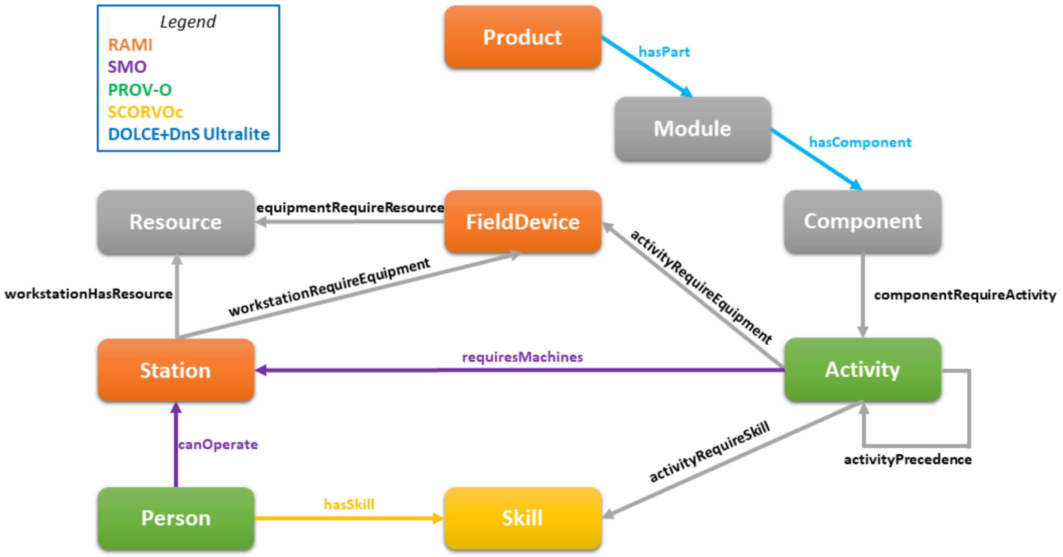


FIGURE I.2.2: The ontology model of a wire harness assembly line, based on the applied vocabulary and namespace elements

TABLE I.2.2: List of the applied Object and Data Properties

Object Properties	**Data Properties**
hasPart	activityTime
hasComponent	activityType
componentRequireActivity	activityTypeName
activityPrecedence	activityTypeRemark
canOperate	activityTypeUnit
requiresMachine	additionalTime
activityRequireSkill	componentType
equipmentRequireResource	zoneOfAssembly
activityRequireEquipment	equipmentName & equipmentCost
workstationHasResource	resourceName & resourceCost
hasSkill	skillName & skillCost
workstationRequireEquipment	moduleName

Properties are denoted by different colours. Furthermore, industry-specific data (described in Section C.1) has been implemented in the ontology as Data Properties, which are listed in Table I.2.2 together with the final applied Object Properties.

The final Protégé implementation of the wire harness manufacturing ontology shown in Figure I.2.3 provides a structural overview of the model in VOWL format. The dark-blue-coloured Classes and Object properties come from prefixes/namespaces, and the Data properties denoted in green are visualized together with their data types. This brings us to the end of the ontology modelling, which can be exported in any RDF format and proceed with data analysis and queries.

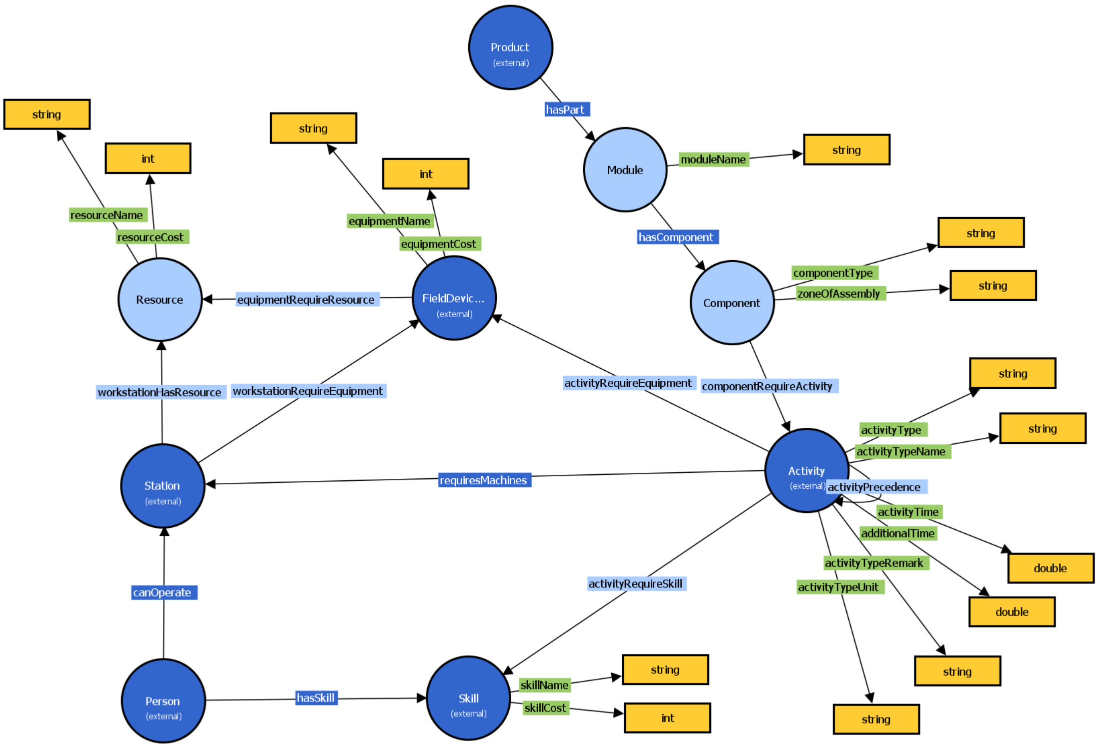


FIGURE I.2.3: The VOWL view of the created ontology of the wire harness manufacturing process

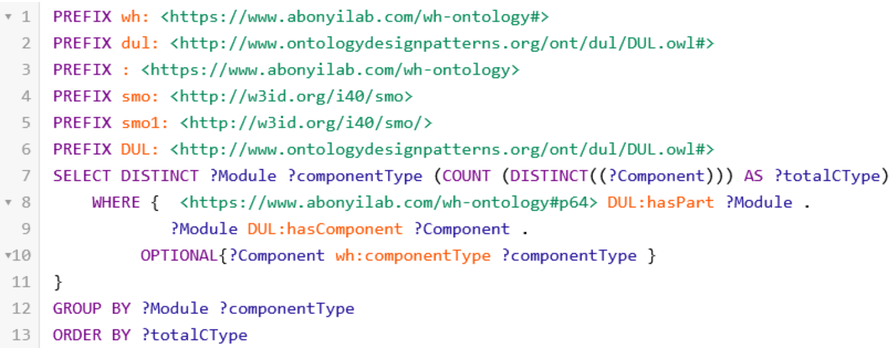
I.2.3 Data queries and evaluation of ontology data

This section describes the creation of the SPARQL queries of the ontology data to analyse the current production state. The collected manufacturing data is based on the wire harness assembly case study presented in Section C.1 of the appendix (Chapter C). The method is evaluated to discover the potential of the line balancing improvements.

In the first case, the scope is, how much unique Component is required for the seven different Modules from the five distinct types of Components as a wire or terminal. Figure I.2.4 shows the SPARQL query to get these Module-Component data.

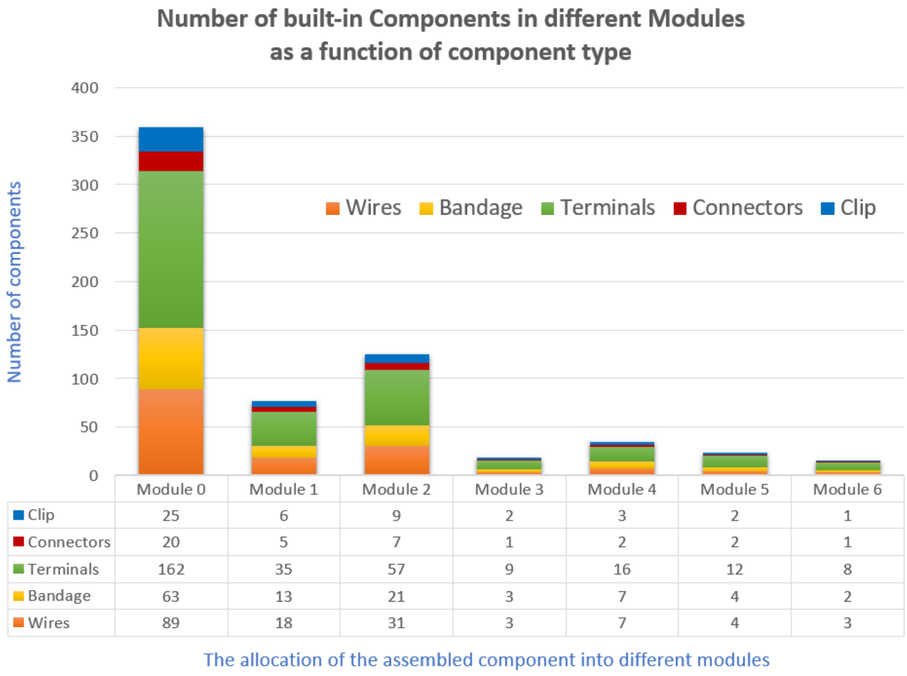
Figure I.2.5 represents the result of the query, and it can be stated in Figure I.2.5, that the most complex wire harness module is m_0, which is the base module, with more than 350 Components, while m_6 has the fewest. It can be seen that the number of different components is evenly distributed in each module, so terminals are the most and connectors are the fewest in every module. The implementation of the Bill of Materials in the ontology (or data model) may yield valuable information, which can support the work of process engineers or designers. The analysis of the assembled components per module is critical toward discovering the relevance of the module to get more precise production scheduling.

In the second case, the most complex product (p_{64}) is investigated. All seven different modules are involved in this wire harness product and the entire assembly process is distributed over ten workstations. Figure I.2.6 describes the query for Workstation-Skill analyses regarding the workstation allocation and skill usage.



```
PREFIX wh: <https://www.abonyilab.com/wh-ontology#>
PREFIX dul: <http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#>
PREFIX : <https://www.abonyilab.com/wh-ontology>
PREFIX smo: <http://w3id.org/i40/smo>
PREFIX smo1: <http://w3id.org/i40/smo/>
PREFIX DUL: <http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#>
SELECT DISTINCT ?Module ?componentType (COUNT (DISTINCT((?Component))) AS ?totalCType)
    WHERE {  <https://www.abonyilab.com/wh-ontology#p64> DUL:hasPart ?Module .
            ?Module DUL:hasComponent ?Component .
         OPTIONAL{?Component wh:componentType ?componentType }
}
GROUP BY ?Module ?componentType
ORDER BY ?totalCType
```

Figure I.2.4: SPARQL query - Module-Component



	Module 0	Module 1	Module 2	Module 3	Module 4	Module 5	Module 6
🟦 Clip	25	6	9	2	3	2	1
🟥 Connectors	20	5	7	1	2	2	1
🟩 Terminals	162	35	57	9	16	12	8
🟨 Bandage	63	13	21	3	7	4	2
🟧 Wires	89	18	31	3	7	4	3

FIGURE I.2.5: Results of the SPARQL query regarding built-in Components in different Modules

Figure I.2.7 illustrates how much built-in Component related activity is assigned to each workstation to assemble this Product. The figure also summarizes the costs required to apply a skill, and the $\sum$ values on the bars represent the total skill costs at each workstation. Based on the used skills, it can be noticed that the $w_4 - w_7$ workstations are similar, which means the activities between these stations can be reallocated without causing additional cost by training. Furthermore, there is a high correlation between skills and types of equipment, so it would not require an additional tool or resource. Considering these, an update or redesign of activity assignments among workstations could reduce the cost of the assembly process.

In the following, the p_1 product is analysed, where only the base module (m_0)

```
PREFIX wh: <https://www.abonyilab.com/wh-ontology#>
PREFIX dul: <http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#>
PREFIX : <https://www.abonyilab.com/wh-ontology>
PREFIX smo: <http://w3id.org/i40/smo>
PREFIX smo1: <http://w3id.org/i40/smo/>
PREFIX DUL: <http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#>
SELECT DISTINCT ?Station ?Skill ?skillName (COUNT (DISTINCT((?Activity))) AS ?totalActivity)
    WHERE {  <https://www.abonyilab.com/wh-ontology#p64> DUL:hasPart ?Module .
            ?Module DUL:hasComponent ?Component .
            ?Component wh:componentRequireActivity ?Activity .
            ?Activity smo1:requiresMachines ?Station .
            ?Activity wh:activityRequireSkill ?Skill .
            ?Activity wh:activityRequireEquipment ?FieldDevice .
        OPTIONAL{?Skill wh:skillName ?skillName }
}
GROUP BY ?Station ?Skill ?skillName
ORDER BY ?totalActivity
```

FIGURE I.2.6: SPARQL query - Workstation-Skill

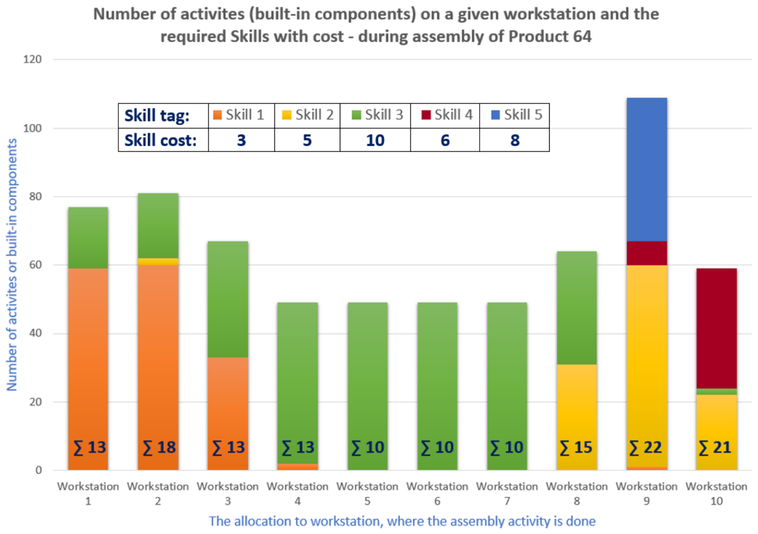
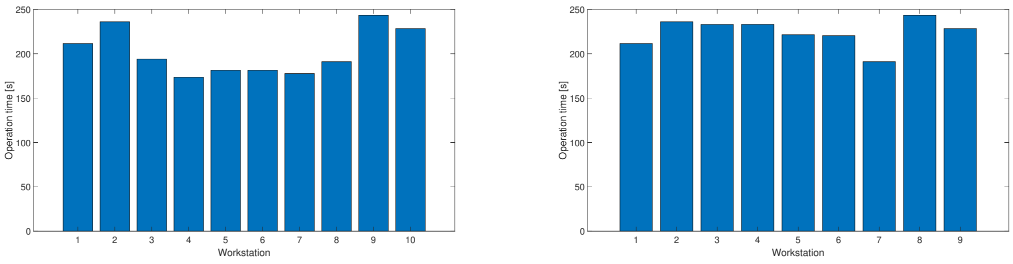


FIGURE I.2.7: Result of the SPARQL query regarding workstation allocation and skill usage during the assembly of Product 64

is assembled because this is the most relevant one (see Figure I.2.5). The line balancing has been analysed to perform further investigations. The part (a) of Figure I.2.8 shows the current line balancing in the case of the p_1 product. It can be noticed that this is not a well-balanced production process. However, the procedure of the applied conveyor line has to be followed. In an open-paced conveyor, the start and the ending stations have more flexibility than the middle ones. Based on that, the operators in the middle stages $(w_3 - w_8)$ are usually planned for lower capacity. Apart from that, it can be also highlighted that the differences are significant nearly a minute between these stations, which is an opportunity to make further analyses to discover the potential of merging these workstations.



(a) The original case

(b) After line balancing

FIGURE I.2.8: The evolution of manufacturing time during assembly of p_1 product before and after line balancing

The (b) part of Figure I.2.8 shows the result of line balancing after reallocation assembly activities related to $w_3 - w_6$ stations and eliminating the w_7 station from the line. Based on the analytics, one workstation (and one operator) from the production line could be eliminated. It is possible to redesign the conveyor line with 9 stations instead of 10. Although there are still gaps among stations, this is more efficient as the starting point. It must be highlighted that only one type of product is in focus, and the open-paced conveyor has a special line balancing rule, as mentioned above. However, the SPARQL-based data queries can make the discovery of communities and critical elements of the production system more efficient. This method can show the possibilities for process engineers to solve the line balancing problem considering all production parameters.

I.2.4 Summary of the ontology-based modeling of a manufacturing process

In this chapter, an ontology development method has been presented with a wire harness assembly-based benchmark. The ontological modeling of a manufacturing process and the data queries and evaluation can be very complex. As a summary, the following list contains an advised strategy to facilitate this process:

- The integration of ISA and IEC standards is important in semantic model-based system development.
- In-depth study of Open Vocabularies can facilitate ontological and semantic modeling.
- There is a need to use and develop industry-specific ontologies and knowledge graphs.
- Data query methods such as SPARQL provide an efficient data processing solution, which can be utilized in semantic networks and knowledge graphs.
- SPARQL queries of the data model can serve as a source for analysing line balancing problems.

This chapter highlighted that human frontline workers on the shop floor have outstanding importance in the assembly industry. Therefore, the following chapter describes the development of a knowledge graph related to the human-centric approach.

Chapter I.3

Knowledge graph-based framework to support human-centered collaborative and ergonomic manufacturing in Industry 5.0

This chapter proposes the Human-Centric Knowledge Graph (HCKG) framework by adapting ontologies and standards that can model the operator-related factors such as monitoring movements, working conditions or collaboration with robots. Furthermore, graph-based data queries, visualization and analytics are also presented in the form of an industrial case study. The main contribution of this work is a knowledge graph-based framework, where the work performed by the operator is of concern, including the evaluation of movements, collaboration with machines, ergonomics and other conditions. Additionally, utilization of the framework is demonstrated in a complex assembly line-based use case, by applying examples of resource allocation and comprehensive support concerning collaboration between the shop-floor workers and ergonomic aspects.

The main goal of this chapter is to propose a knowledge-graph framework for the modeling, supporting, and scheduling of the operator, where in addition to efficient data collection, the work of the operator can be facilitated by the use of a KG and the implementation of Industry 5.0 technologies becomes possible.

This main contribution is proposed in Section I.3.1, where the building elements of the HCKG design concept are defined. The applied industrial case study is discussed in the Appendix (Chapter C) in Section C.2. In Section I.3.2 the different human-robot collaboration scenarios are summarized. Then, Section I.3.3 proposes the relevant human-centric performance indicators. The utilized methodologies and software tools are presented in Section I.3.4. Additionally, Section I.3.5 describes the development procedure of the use case-specific knowledge graph. In Section I.3.6, the results of the KG-based analytics are presented. Finally, in Section I.3.7 the contributions are summarized.

I.3.1 Human-centered knowledge graph towards collaboration in manufacturing

This section discusses the main contribution of this chapter, the human-centered knowledge graph (HCKG) design concept. Subsection I.3.1.1 discussed the manufacturing operations management related activity model, then subsection I.3.1.2 presents a monitoring system concept. Finally, in subsection I.3.1.3 the structure of the HCKG concept is presented.

Semantic technologies such as ontologies, graph databases, semantic analytics and reasoning provide an efficient way to process a large amount of data from various sources, as the entire data set becomes transparent and accessible [180, 181]. In order to improve the working conditions of operators, different monitoring systems can be used such as sensor networks, which can follow where the operator goes and their physical conditions [182, 183]. Semantic networks and graph-based analytics are recommended to handle the process information using linked data features.

Additionally, Industry 5.0 technologies pioneering solutions to provide safer and more comfortable working conditions while ensuring access to technologies that enable automation and increase productivity. The key enabling technologies of Industry 5.0 are cobots, 6G and beyond, digital twins, blockchains, the Internet of Every Thing, big data analytics, edge computing and artificial intelligence [35]. For example, a service or assembly procedure can be facilitated by AR or production development scenarios modeled with a digital twin before redesigning the shop floor. The novelties of Industry 5.0 research are recommended to facilitate human-machine collaboration such as AR-aided assembly or creating human digital twins for optimization purposes.

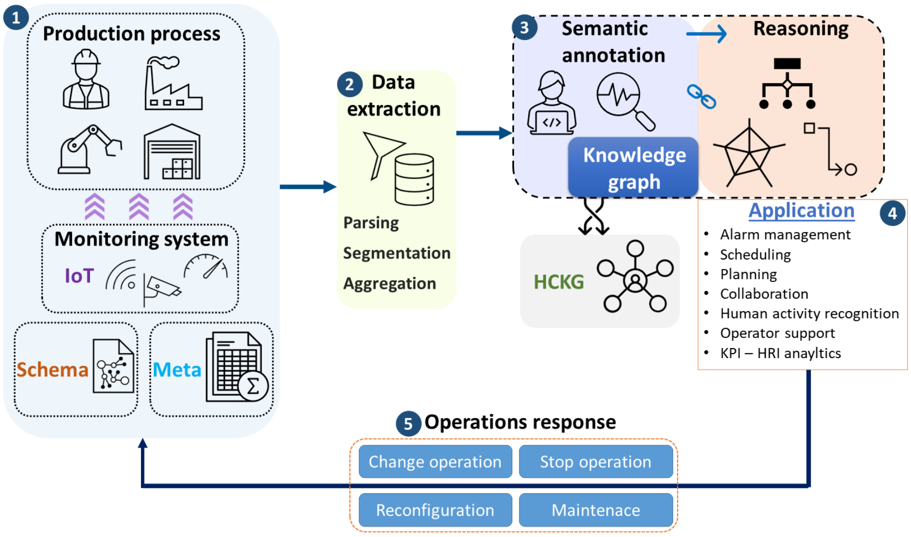


FIGURE I.3.1: Integration of the HCKG design concept into a production process, using five segments

In general, the HCKG aims to offer effective human-machine collaboration, resilience, agility and improved working conditions for the operator. The knowledge graph includes the monitored information about the activities of the operator, the environment as well as all robots and assets which are present in the manufacturing space. By analysing the related knowledge graph data, the collaboration can be improved and work instructions tailored to the workers, moreover, any changes that may occur can be handled adaptively.

First, Figure I.3.1 shows a general integration method of the HCKG concept, which also serves as a graphical abstract. In the first segment, the *Production process* element represents the complex production environment, containing all human-machine resources, processes, activities and interactions. The *Monitoring system* element interacts with the production process and collects historical and live data with sensors and *IoT* devices. Additionally, the *Schema* element provides the semantic tools to get contextualized data model, and the *Meta* element contains the meta information such as industry standards to ensure re-usability. The first segment contains several structured, and unstructured data sources, that have to be pre-processed. Therefore, the second segment contains the *Data extraction* element, which includes processes, such as parsing, segmentation or aggregation of data. The goal of data extraction is to identify and extract relevant data from unstructured or semi-structured data sources of the first segment, to convert it into a structured format that can be analyzed and used in optimization.

The third segment contains the *Semantic annotation* and the *Reasoning* elements, which utilize semantic modeling and data analytics on a complex knowledge graph. The *Semantic annotation* block creates the *Knowledge graph*, using the schema, meta information and extracted data. Semantic annotation involves adding metadata, standardized labels, or tags to the entities and relationships in the knowledge graph, such as industry-specific terminology or concepts from a particular domain. It allows the application to more accurately identify and categorize different entities in the knowledge graph, by providing additional context and allowing for more accurate categorization and identification of entities in the data model. The *HCKG* block stands for the human-centered knowledge graph element of the built semantic network, which can be the entire KG or only the shop-floor worker-related part of it, depending on the use case.

The *Reasoning* element provides the enriched semantic information for the following, fourth segment, which is the *Application*. The reasoning process is based on the idea that the relationships and connections between different entities in the knowledge graph can be used to draw logical conclusions and make new predictions. In the context of analytics and optimization, semantic reasoning can be used to identify patterns, correlations, and causal relationships between different entities in the KG. By applying semantic reasoning, the application can identify patterns and correlations between these different data points, such as identifying which machines are most likely can cause time delays on a specific production line. By reasoning over the knowledge graph, the application can identify the optimal sequence of steps in the production process that will minimize waste and maximize efficiency. Integrated human-centered knowledge graph applications can be utilized i.e. for the following tasks: alarm management, scheduling of operations or manpower, monitoring, and optimization of human-machine collaboration, human activity recognition or analytics of performance metrics.

The result of the application besides analytical results, can be the *Operations response* (fifth segment), which is bypassed to the *Production process* element. A response can be e.g. the following production orders: Change operation, Stop operation, Reconfiguration or Maintenance.

I.3.1.1 Manufacturing operations management

This subsection discusses an extended MOM (Manufacturing Operations Management) activity model, which is visualized in Figure I.3.2, where the elements can be considered according to the time they occur when the work is executed.

For a more comprehensive discussion of the problem, an extended MOM (Manufacturing Operations Management) activity model has been investigated, based on [184], which is visualized in Figure I.3.2, where the elements can be considered according to the time they occur when the work is executed. The temporal view of the generic activity model as Pre-, Actual-, Post-Work and Reference data is also highlighted [185]. Furthermore, the extension modules of the standard activity model of MOM [184] are visualized at the bottom in brown.

The MOM approach aims to show in detail the mechanisms associated with the operator during a general manufacturing activity, moreover, focuses on the properties of the added monitoring and support framework elements. Since the generic activity model is divided into four parts based on the temporal view, which are highlighted with green labels on the figure the model is analysed and discussed in a similar manner.

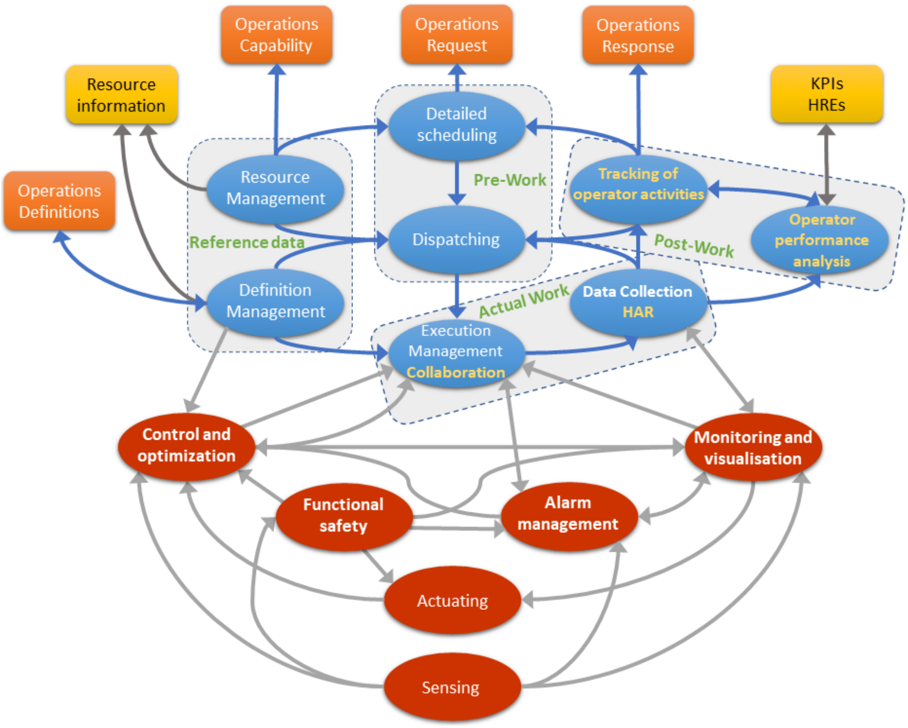


FIGURE I.3.2: Activity model of manufacturing operations management with an operator-centric view

The *Reference data* contains all the information about specific operators such as capabilities, skills and experience in certain fields. The *Resource* and *Definition Management* blocks of the MOM store aggregate this information and determine base data for the following work sections of the model. As an extension to the reference data section, the *Control and optimization* block is recommended, where machine learning [186, 187] or artificial intelligence-based solutions [188] can improve the ongoing production processes.

The second part in Figure I.3.2 is the *Pre-work*, where the *Detailed scheduling* is utilized based on the *Operations Request*, moreover, the *Dispatching* is performed. These activities ensure that all operators receive adequate work instructions, scheduling and are optimally allocated.

The *Actual-work* section of the MOM describes the activities which are happening at present and are controlled by *Execution Management* while *Data Collection* is in progress. Some human-centered aspects are added (denoted in yellow text), such as *Collaboration* or utilization of Human Activity Recognition (*HAR*) sensor technologies. Given that real-time operator support should be reinforced, *Alarm management*, *Monitoring and visualization* are added as extension elements. An alarm management system [189] can prioritise, group as well as classify the alerts and event notifications used in the Supervisory Control And Data Acquisition (SCADA) system, improving performance and monitoring levels of safety. A smart monitoring system can collect data concerning various manufacturing objects such as the temperature, noise or vibrations and obtain them in real-time to provide a graphical visualization and alerts when an abnormality occurs [190]. For example, a high-level visualization technique can be based on augmented reality that assists the operator by providing information from the digital twin [191].

Finally, during the *Post-work* period of the activity, the *Tracking of the operator activities* is performed to obtain an *Operations Response* for the MOM. Furthermore, the *Operator performance analysis* is utilized, which is the source of the KPIs (key performance indicators) as well as HRE (human resource effectiveness), key elements in the KG to enable resilient and agile conditions for the operators.

The briefly discussed extension modules of the activity model are interconnected to the KG with semantic technologies. The emerging smart cyber-physical systems create the framework where each human and machine segment of the complex manufacturing system is appropriately monitored and the information systems are interoperable [192, 193].

The essential parts of the extended MOM model, from the perspective of the shop-floor workers, are the *Operator Performance analysis*, *HAR* and *Monitoring*, which are key to KPIs and metrics analysis. A comprehensive analysis of operator performance can facilitate competence-based matching and the formation of competence islands. Since the demand for reconfigurable production lines is increasing, the static assembly lines may be replaced by autonomous workplaces known as competence islands, where mobile robots move between these islands. Additionally, the competence islands need to be equipped with collaborative robots capable of working safely and reliably with operators [194]. Another related feature is competence-based matching, where comparisons between work system requirements and the competences of employees are performed. The competence-based description of employees plays an important role in reconfigurable manufacturing systems [195]. Additionally, another paper studied the semantic modeling as well as analysis of task and learning profiles in terms of human-machine collaboration [196] to establish a qualitative and quantitative methodology for the optimal selection of a competent jobholder profile. The so-called Vector of Competence and Autonomy (VCA) is designed to identify the extent of human-machine collaboration. Being highly important, in the following section, the monitoring perspective of manufacturing operations is discussed in more detail.

I.3.1.2 Monitoring system concept

This subsection presents the theoretical structure of a conceptual monitoring system (see in Figure I.3.3). Three different building elements are defined, namely *Production process*, *Sensor* and *Monitoring and supervision.* Furthermore, the evaluation segment is represented in the connected *Manufacturing operation and management* element.

In the *Sensor* part, different characteristics and conditions are highlighted such as vibrations, locations or noises. These environmental, position or behavioral factors are measured with regard to the location of *Machines*, *Operator* or *Automated production*, which can contain a collaboration of different actors. Furthermore, some of the monitored factors are highlighted in blue in this figure. The evaluating system is able to calculate the KPI measures from the monitoring system data

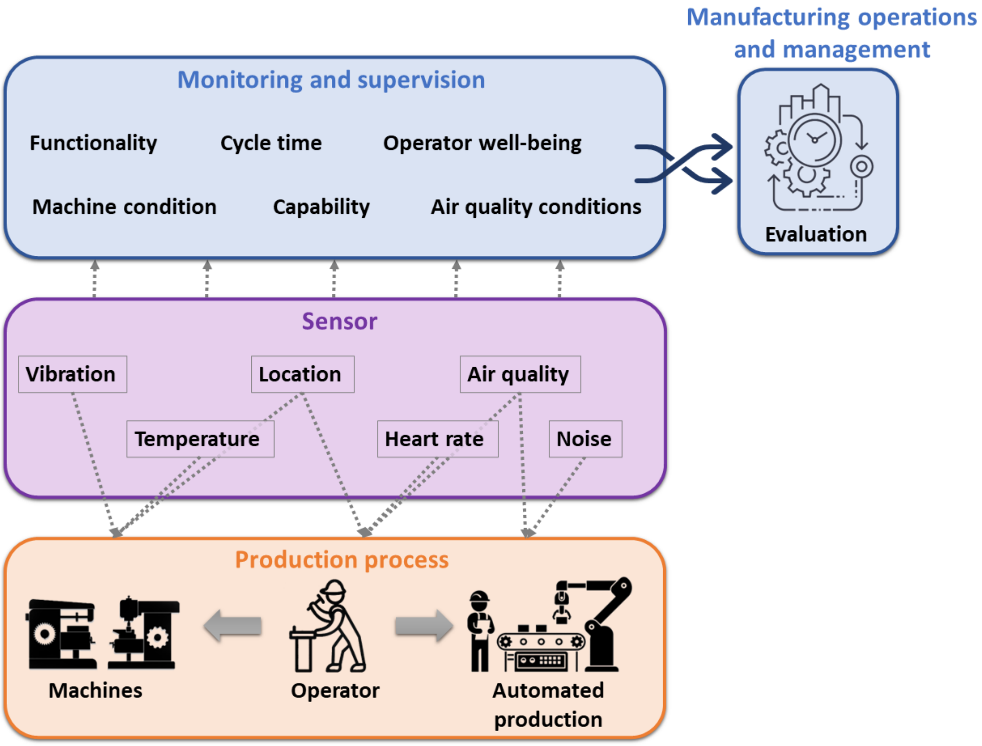


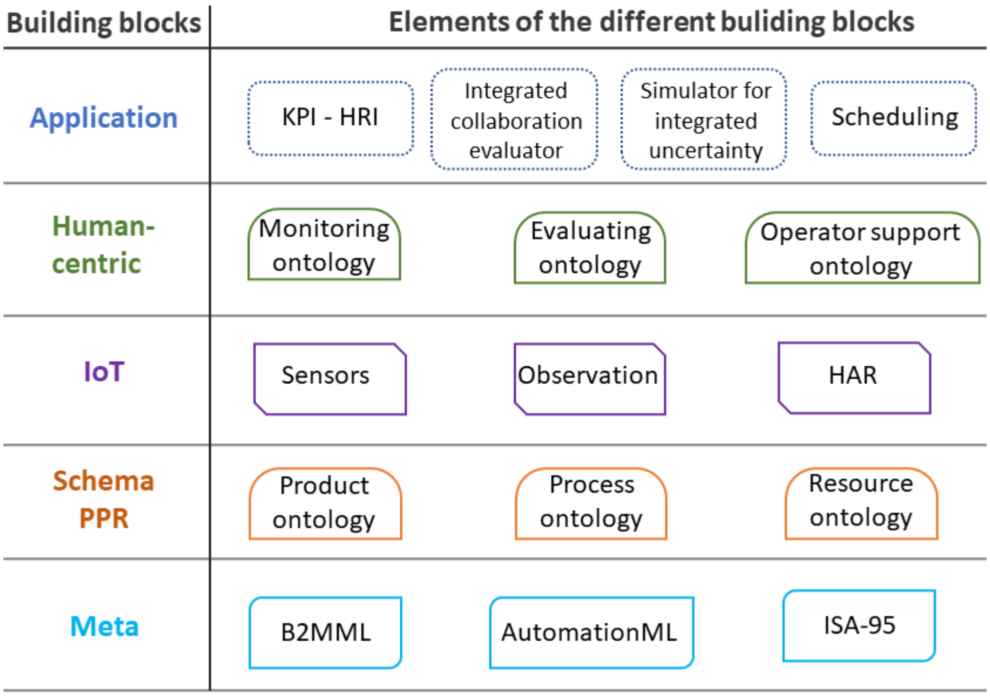
FIGURE I.3.3: Monitoring system concept

about the operators as well as provide real-time functionality information about the production line. Additionally, given uncertainties in the measurements and possible inaccuracies in sensor databases, adequate data models are required to represent these factors [197].

I.3.1.3 Design structure of the HCKG concept

This subsection summarizes the methodology, and provides an overview of the proposed development framework in a block structure (see in Figure I.3.4), where the aim is to position the human-centric KG block in a complex industrial environment. The framework consists of five different blocks (or segments), starting with the meta-data sources of a business or industry network and finishing with the application, where the information is utilized to create value.

Starting from the bottom, the *Meta block* contains all the data necessary to describe the business processes and the describable factors of a facility, e.g. material or information flows. Markup languages and standards, e.g. B2MML (Business To Manufacturing Markup Language), AutomationML or ISA-95, give the initial structure for addressing as well as managing the variety of data sources and



Building blocks	Elements of the different buliding blocks			
Application	KPI - HRI	Integrated collaboration evaluator	Simulator for integrated uncertainty	Scheduling
Human-centric	Monitoring ontology	Evaluating ontology	Operator support ontology	
IoT	Sensors	Observation	HAR	
Schema PPR	Product ontology	Process ontology	Resource ontology	
Meta	B2MML	AutomationML	ISA-95	

Figure I.3.4: Theoretical structure of the proposed human-centered knowledge graph-based design concept

processes in a complex network. Extension of already existing standards such as ISA-95 is recommended. An essential aspect of industrial development is the utilization of standardized models, which facilitates more efficient integration of a new design concept into a production system as well as expansion of existing methodologies, making the learning period of technical features more dynamic.

The second is the *Schema and PPR block*, which stands for the three descriptional ontologies at an Industry 4.0 facility. The Product, Process and Resource ontologies can describe the entire network in a semantic form. Different assets, physical or human characteristics, attributes, and concrete values are modeled as ontology axioms (individuals), which are categorised into classes. Additionally, semantic properties, rules and queries make the interoperability and description of connections possible, e.g. the capabilities of actors, sequence of manufacturing activities or allocation of resources.

The *IoT block* contains the monitoring devices and sensors to perform observations as well as human activity recognition (HAR) that are required sources for the higher, human-centric block - plays an intermediate role. Additionally, IoT devices form a complex system, which requires them to be managed in a separate segment, as the variety of smart devices and sensors is diverse.

The *Human-centric block* consists of the Monitoring, Evaluating and Operator-support ontologies, which aim to collect as well as process all the applicable information about the production process, collaboration, human activities or working conditions on the shop floor. The main goal of this block is to keep the operator in the loop and support in ergonomic, collaboration and other aspects. In a human-robot collaborative environment, feedback can be important not only from monitoring or machine side, but also from the "human" side, focusing on what are the real ergonomic characteristics, process parameters and other feedback from the operator. Operators of the shopfloor may provide valuable information for the *Operations Response* of the MOM, which shall be bypassed into the semantic-based data management, aims to support the CI/CD (Continuous Integration and Continuous Delivery) best practice.

Finally, the *Application block* contains all the information value that HCKG can provide and utilize for scheduling, resource allocation, improving KPI and HRI factors, evaluating collaboration aspects or performing simulations. The end user, who might be a process engineer, shop-floor workers, or the production responsible are only concerned with this segment, as it delivers the final result of the semantic-based analysis. The application block can facilitate the study of integrated uncertainty using simulations and evaluate the collaboration or business processes. Additionally, scheduling or allocations can be optimized based on the resulting performance metrics.

Additional topics, such as the cyber-security issues of large-scale infrastructure which are not addressed here, will probably remain one of the main issues for years to come.

After presenting the HCKG design concept, the following section discusses briefly the relevant human-robot collaboration scenarios.

I.3.2 Human-robot collaboration scenarios

This section discusses the different types of workstations and collaboration scenarios, which are important in the scope of the applied case study.

Different types of workstations can be distinguished depending on the allocated human or robot workforce. Three types of workstations, depending on human or robot actors, are presented in Figure I.3.5 [27]. In the presented case study, all three types can be found. *Crimping stations 1* and *3* are manual workstations, while *Crimping stations 2* and *4* are automatic ones. The case study contains four collaborative workstations, namely *Assembly stations 1-4*.

To obtain a more detailed case study, in the case of a collaborative workstation, a further classification is made depending on the interaction between the human and robot actor in terms of work. Three different types of collaboration are shown in Figure I.3.6 [198, 199]:

1. **Separate work**
   Human and robot tasks are kept apart and they do not share workspaces, tools or workpieces.
2. **Sequential collaboration**
   Although the human and robot actors are in a shared process flow of a workpiece, tasks are completed in succession. The workspaces, tools and workpieces may be shared, but the tasks are strictly serialized such that any sharing is temporally separated.
3. **Simultaneous collaboration**
   The human and robot tasks are executed concurrently, moreover, may involve working on different parts of the same workpiece, but are focused on achieving separate task goals.
4. **Supportive collaboration**
   Humans and robots work together and on the same workpiece to complete a common task.

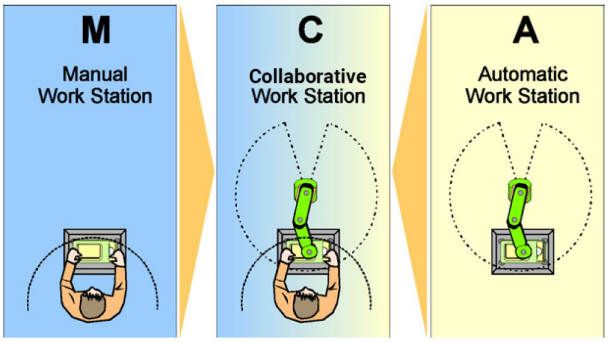


FIGURE I.3.5: Manual, automatic and collaborative types of workstations [27]

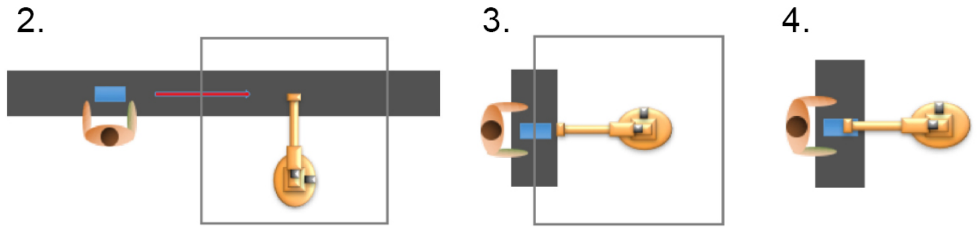


Figure I.3.6: Sequential (2.), simultaneous (3.) and supportive (4.) types of human-robot collaborations [198]

In the presented wire harness assembly-based case study (Appendix C.2), collaboration types 3. and 4. are discussed. A concrete example of the simultaneous and supportive types of collaboration is given in Figure I.3.7. In the case of *Result* 28 and 31, since the human and the robot actors perform the same types of activities on the same product, these are supportive collaborations performed to achieve the same assembly result. On the other hand, *Result* 29 and 30 are related to different types of activities, so the human and robot actors work on the same product at the same time but for different goals.

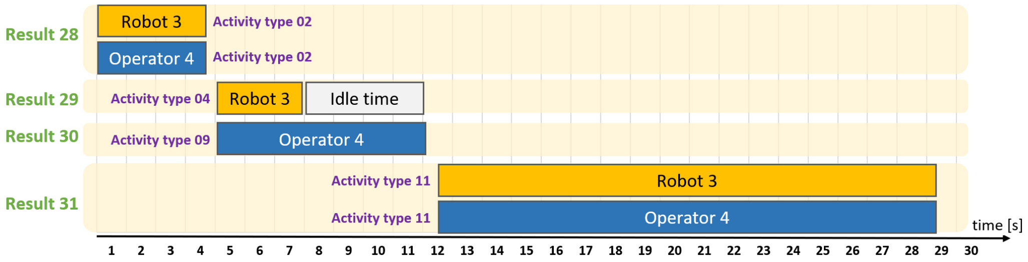


Figure I.3.7: Gantt chart of collaboration scenarios

The following section introduces the key performance indicators, which are relevant with regard to the case study of a human-centric knowledge graph.

I.3.3 Performance indicators of collaborative manufacturing

This section provides an overview of the most relevant key performance indicators in the form of a human-centric, ergonomic and human-robot collaboration.

In the presented framework, one of the operator monitoring features druing human activity recognition is to analyse the movements with an ergonomic point of view

[122]. Using image processing techniques the skeleton model of the worker can be analyses, and adding taxonomies, it can be broken down into parts, movements, elementary postures, for which ergonomic models can be assigned [200].

Besides the MOM-related metrics in the case of Industry 5.0, designing a human-centered smart environment requires even more factors to prioritise human well-being while maintaining production performance. Defining the appropriate evaluation factors for human-robot-machine-collaborations and the mentioned ergonomic factors of factory workers is in high demand [201]. A comprehensive framework for the evaluation of human-machine interfaces (HMI) and human-robot interactions (HRI) in collaborative manufacturing applications is needed [198]. An outstanding systematic review [16] identified and categorised the measures, metrics and quality factors adopted or applied in the HRI literature using a systematic approach. The categories of metrics with regard to aspects of Industry 5.0 research are the following [16]: Physical ergonomics (Safety, Physical workload, Workplace design), Cognitive ergonomics (Mental workload, Awareness), Performance (Efficiency, Effectiveness) and Satisfaction/Hedonomics with regard to user experience (Emotional responses, Acceptance, Attitudes, Trust).

Although this work does not give a systematic overview of this topic, a paper evaluating the quality of human-robot interaction [16] has been partly adapted. Additionally, from a semantic technology point of view, a publication of ontology-driven KPI metamodelling [15] has also been considered in this case study.

The human-centric KPIs advised for this case study are summarized in Table I.3.1, using six different categories, namely Time behaviour, Physical measures, HR physical measures, Efficiency, Effectiveness and Ergonomics. Furthermore, in the second column, the Operator 4.0 types [37] have been added to these KPIs, representing the potential to support the development of human-automation symbiosis.

The utilized methods and software tools for KG creation, mapping and analysis are presented in the following section.

Table I.3.1: The categorised human-centric KPIs for the case study

KPI description	**Operator 4.0 type**
Time behaviour category	
Average time to complete task	Analytical operator
Collaboration time - Type-3 and Type-4	Collaborative operator
Functional delays	Analytical operator
Human operation time	Analytical operator
Interaction time	Collaborative operator
Response time	Collaborative operator
Robot action time	Collaborative operator
Robot functional delay	Collaborative operator
Robot operation time	Collaborative operator
Task completion time	Analytical operator
Total assembly time	Analytical operator
Total operation time	Analytical operator
Physiological measures category	
Biosignals (temperature, tactile, etc.)	Healthy operator
Ergonomics improvement	Healthy operator
Muscle activity	Healthy operator
Ocular behavior	Healthy operator
HR physical measures category	
Avg./min. length between a human hand and robot hand	Collaborative operator
Human-robot distance	Collaborative operator
Efficiency category	
Availability	Collaborative operator
Average robot velocity	Collaborative operator
Concurrent activity	Collaborative operator
Degree of collaboration	Collaborative operator
Layout efficiency	Analytical operator
Effectiveness category	
Accuracy	Analytical operator
Interaction accuracy	Collaborative operator
Level of assignment	Collaborative operator
Level of interaction	Collaborative operator
Overall equipment effectiveness	Analytical operator
Real-time human fault	Analytical operator
Real-time robot fault	Collaborative operator
Ergonomics - environmental category	
Environmental condition - noise	Healthy operator
Environmental condition - humidity	Healthy operator
Environmental condition - temperature	Healthy operator
Environmental condition - gases	Healthy operator

I.3.4 Applied methodologies and software tools of the specific knowledge graph

This section briefly discusses the applied development methods and software tools utilized in sections I.3.5 and I.3.6.

Several processing stages of a data pipeline based on a study [50], which aims to create KGs for the automation industry, are presented in Figure I.3.8. Additionally, an end-to-end digital twin pipeline [202] has been considered.

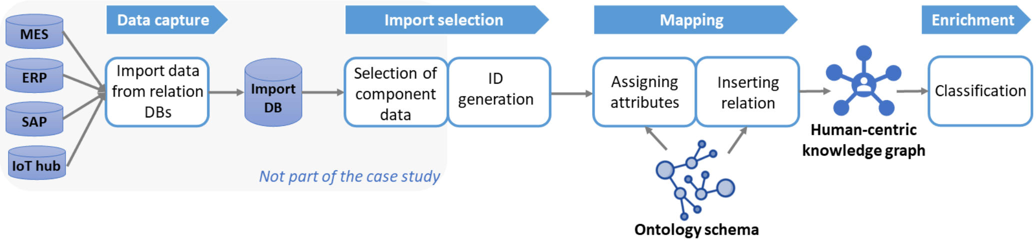


FIGURE I.3.8: Knowledge graph pipeline based on [50]

The data capture and import selection parts of the pipeline are beyond the scope of this chapter. Only KG, ontology creation, data queries, mapping, and data enrichment and visualization are discussed. The phases, applied methods and different software stages of the presented industrial case study are shown in Figure I.3.9.

Firstly, the sub-ontologies and entire KG were developed using Protégé [170] before the TTL file was processed in a Python environment using Pyvis (a Python library for visualizing networks) [203] and KGlab [204, 205]. The data imported into the ontology skeleton as well as the creation of axioms and properties can be made either with Protégé or KGlab in Python. For each data query, the SPARQL language was utilized [206], for that Pyvis offers graphical visualization as well.

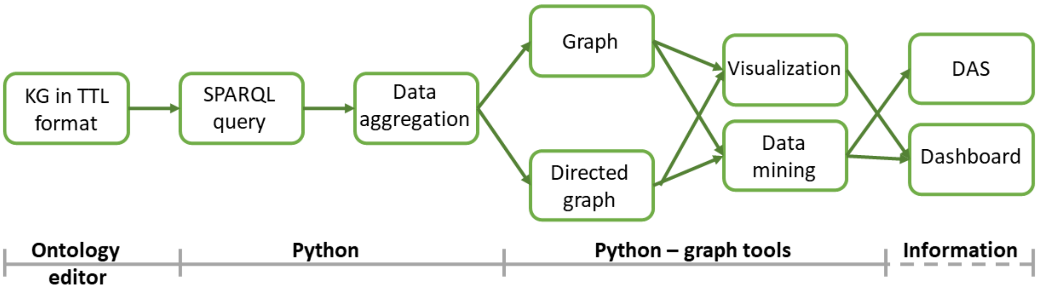


FIGURE I.3.9: The steps of the applied method

After mapping the semantic data, it was further aggregated in Python, in order to obtain data-enriched graphs for analysis. The graph-based visualization of KG data can also be normal, directed or a hypergraph. Finally, as a concept (denoted by a dashed line in Figure I.3.9), the key information, created charts, statements or messages can be displayed on dashboards and DAS devices or fulfill any other elements of the application layer with data, as previously presented in Figure I.3.4.

The description of human-centric KG creation for the industry-specific case study is discussed in the following section.

I.3.5 Development of the use case-specific human-centered knowledge graph

The development of the case study-specific KG, which aims to demonstrate the proposed HCKG concept, is described in this section.

The main development framework is applied, which has been discussed earlier in Figures I.3.1 and I.3.4 of Section I.3.1. Figure I.3.10 shows a part of the developed KG, without the different data properties of the ontology classes. The applied case study is presented in the Appendix C.2 of Chapter C. A more detailed structural diagram of the KG can be found in Figure C.5 of Appendix C.3. As the KG consists of several sub-ontologies, the structural diagram is also divided into six groups of ontology classes in Figure I.3.10. Additionally, the object properties, in the form of relations within classes, are labeled on the arrows. The names of the ontology classes contain prefixes, which show the adapted namespaces from other industry-specific ontologies. These prefixes and the applied ontologies are summarized in the following list:

- **smo - Smart Manufacturing Ontology** [207]
  An ontology to model I4.0 production lines and smart factories based on RAMI 4.0. It highlights the sequence of processes and machines required for a produced workpiece.
- **SOSA - Sensor, Observation, Sample, and Actuator ontology** [98]
  For modeling the interactions between the entities involved in terms of observation, actuation and sampling. Together with SSN (Semantic Sensor

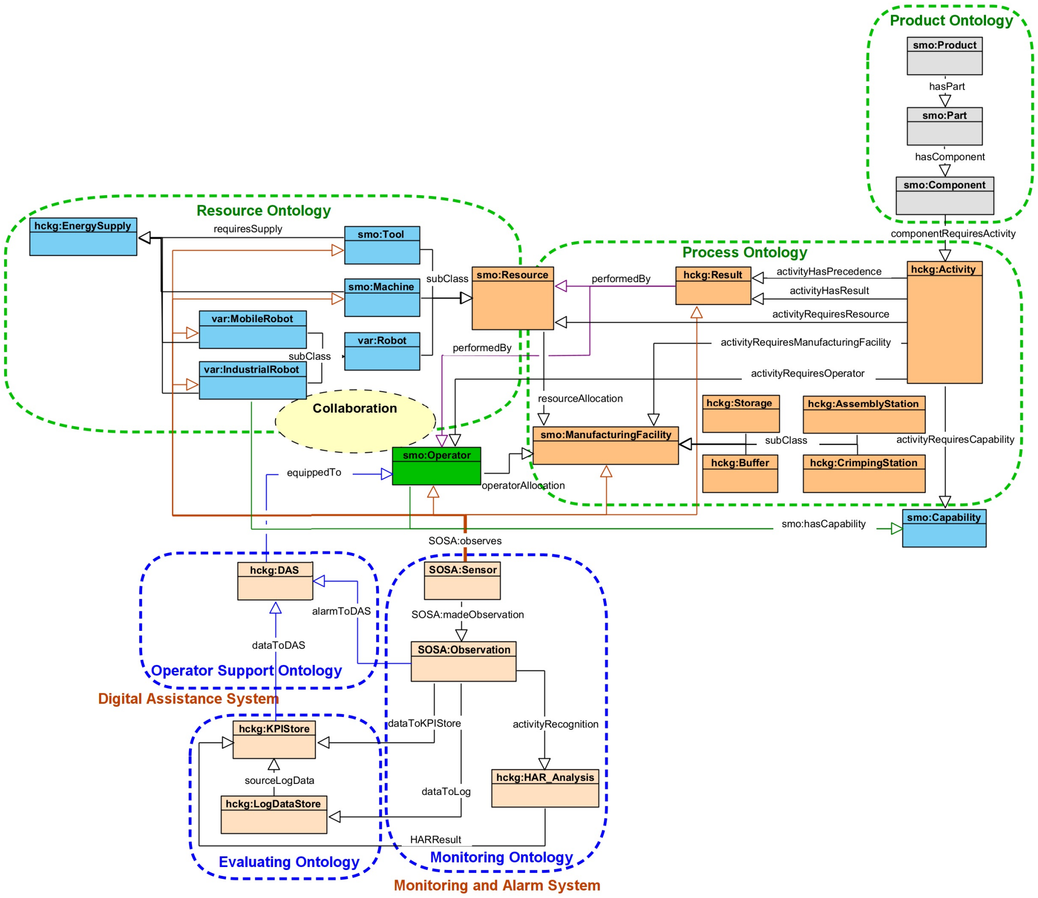


FIGURE I.3.10: Partial structural diagram of the developed wire harness assembly-specific knowledge graph

Network), can be used to describe sensors and their observations, the involved procedures, the studied features of interest, the samples used to do so, the feature's properties being observed or sampled, as well as actuators and the activities they trigger [208].

- **var ontology** [78]

  A core ontology for data exchange in a semantic-oriented framework to support adaptive, interactive, assistive and collaborative assembly workplaces.

- **hckg - Human-centric knowledge graph**

  The authors created a set of classes and properties to model the wire harness assembly-based case study semantically.

The *Product ontology* contains three classes, namely *Product*, *Part* and *Component*. Since this aspect of the wire harness assembly was discussed more in-depth in the previous Chapter I.2, the complexity of this field is not examined here.

The *Process ontology* consists of the following classes: *Activity*, *Result* and *ManufacturingFacility*, which is comprised of other sub classes such as *Storage*, *Buffer*, *AssemblyStation*, *CrimpingStation* and *Capability*.

The main class of *Resource ontology* is the *Resource*, which consists of several sub classes, that is, *Tool*, *Machine* and *Robot*. The *Robot* class is divided even further into *MobileRobot* and *IndustrialRobot*. Furthermore, the *EnergySupply* class is also involved in Resource ontology.

Given that the *Operator* class is the main element of the human-centric KG, it is denoted in green in the middle of the KG structure in Figure I.3.10. Six different object properties are linked to the Operator class, which semantically describes the processes and effects in connection with the personnel on the shop floor.

The *Monitoring ontology* consists of three classes, namely *Sensor*, *Observation* and *HAR_Analysis*. The semantic model of sensor devices as well as their measurements, observation and human activity recognition are stored in this ontology. The *Evaluating ontology* is designed to manage the data originating from the previous three classes and consists of two classes, that is, *KPIStore* and *LogDataStore*. Finally, in the *Operator support ontology*, the *DAS* class describes the digital assistance system.

As the *Operator* and *Activity* classes can be regarded as key classes of the KG, Tables I.3.2 and I.3.3 describe the related object properties.

Table I.3.2: Object properties of the Activity class

hckg:Activity	
componentRequiresActivity	Connects individuals from the Component and Activity classes as well as provides information about the required activity to assemble a specific component on the wire harness.
activityHasPrecedence	Since the assembly procedure requires a specific sequence, certain activities must be finished before another can be started. This is known as the precedence criteria.
activityHasResult	Describes the intended result of a particular activity. In the case of collaboration, several activity individuals may be connected to the same result individual.
activityRequiresResource	Interlinks Tool, Machine or Robot individuals to an activity as a resource requirement.
activityRequiresManufacturingFacility	Workstation requirement of an activity. Connects activity individuals with the ManufacturingFacility individuals such as Storage, Buffer, AssemblyStation or CrimpingStation.
activityRequiresOperator	Connects operator individuals to an activity as a personnel requirement.
activityRequiresCapability	Describes the capability requirement of a specific assembly activity, which has to be conducted by an Operator or IndustrialRobot.

Table I.3.3: Object properties of the Operator class

smo:Operator	
activityRequiresOperator	It provides information about a certain operator involved in certain activities.
operatorAllocation	Semantically connects operators with ManufacturingFacility individuals such as Storage, Buffer, AssemblyStation or CrimpingStation. It provides information about where the operator performs his/her work.
performedBy	Connects Results with Operators and shows which operator was involved in which result(s).
equippedTo	Describes the usage of Digital Assistance System devices by operators.
SOSA:observes	Semantically connects sensor individuals with operators and shows how the personnel are monitored.
smo:hasCapability	Shows which capabilities require a specific operator.

After creating the use case-specific knowledge graph and importing the required data for the semantic network, the next step is to form queries and analyse the results. Therefore, the utilized examples of KG-based analytics are discussed in the following section.

I.3.6 Discussion on KG-based analytics of the use case

This section presents the utilization of visual analytics tools in the resulting industrial case study-related KG. Ontology-compatible queries, data aggregations and several graph visualizations are presented to facilitate human-centered process analysis.

First, the graph visualization of the entire KG of the wire harness assembly-based case study is shown in Figure I.3.11. This semantic representation offers a visual verification of the manufacturing process. The entire network is visualized on the left-hand side containing each property and individual of the KG, while a minor detail is presented on the right-hand side of Figure I.3.11. The orange node represents equipment $E5$ and some of the connecting data properties such as *locationID* (18-4), *equipmentCondition* (86), *equipmentID* ($E5$), *equipmentName* (*ScrewdriverC*) and *equipmentType* (*Screwdriver*).

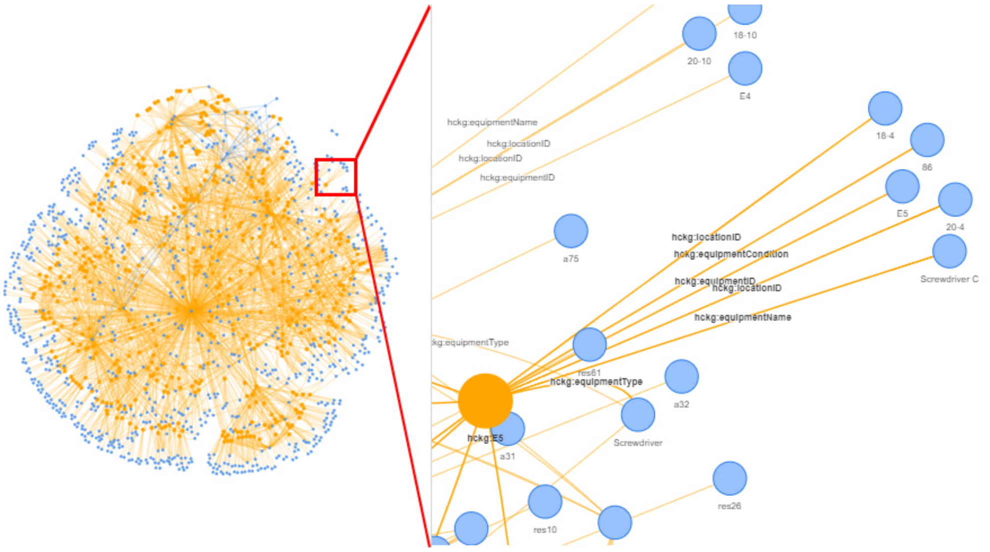


Figure I.3.11: Visualization of the entire knowledge graph of this case study (on the left-hand side) and some of the data properties of equipment $E5$ (on the right-hand side)

The first example of SPARQL [206] query-based data mapping is presented in Figure I.3.12. The detailed SPARQL query can be found in Figure I.3.13. On the left-hand side of Figure I.3.12, the graph visualization of the query can be seen, where four different rules are defined to achieve the desired result. This example is looking for *RobotAsset*s that have an *IndustrialRobot* type, moreover, aims to list three corresponding data items, namely the *Location*, *EnergySupply* and *ManufacturingFacility*. A graph of the query result is shown on the right-hand side of Figure I.3.12. The *IndustrialRobot*-type robot assets are presented as orange nodes, each of which requires an *EnergySupply* called *g2* (Electricity). Each *IndustrialRobot* node is connected to the relevant node of the workstation (*ManufacturingFacility*), which are different Assembly stations denoted in purple in these cases. Finally, the location data properties of the robots are labeled with blue nodes (the robots require two zones on the shop floor). This type of visual analytics can support the investigation of dependencies in the case of specific assets.

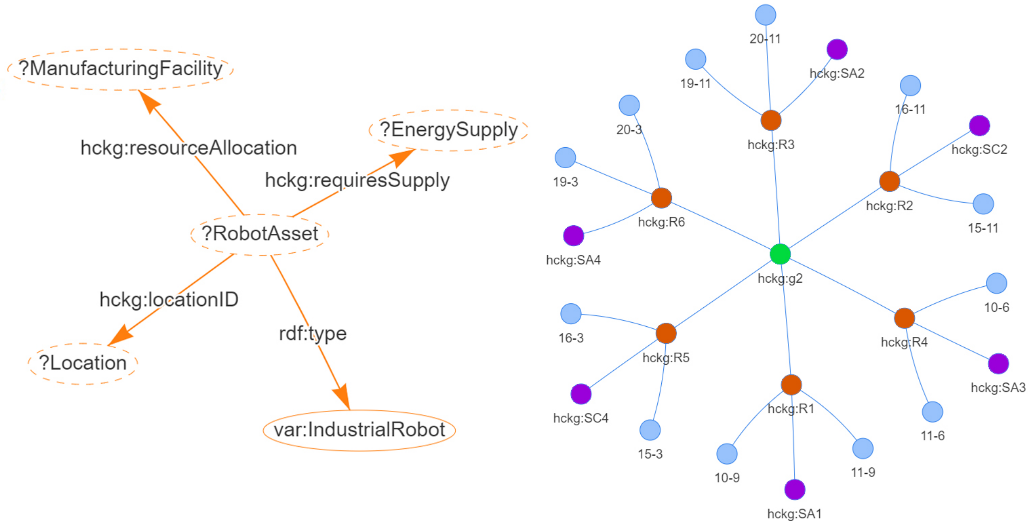


FIGURE I.3.12: Visualization of the *RobotAsset* query (on the left-hand side) and the graph visualization of the result (on the right-hand side)

```
PREFIX hckg: <https://www.abonyilab.com/wh-ontology#>
PREFIX var: <http://www.satisfactory-project.eu/satisfactory/var#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?RobotAsset ?ManufacturingFacility ?Location ?EnergySupply
    WHERE {
        ?RobotAsset hckg:resourceAllocation ?ManufacturingFacility .
        ?RobotAsset hckg:locationID ?Location .
        ?RobotAsset hckg:requiresSupply ?EnergySupply .
        ?RobotAsset rdf:type var:IndustrialRobot
    }
ORDER BY DESC(?RobotAsset)
```

FIGURE I.3.13: SPARQL query - RobotAsset

The query and resulting graph of *Actors* (operators or robots) as well as the *Capability* individuals with whom they are connected are presented in Figure I.3.14. The detailed query in SPARQL can be found in Figure I.3.15. This example can serve as an visual analysis of the manufacturing capability. It can be seen in Figure I.3.14 that capability $C8$ (AGV loading/unloading) is possessed by most actors. Additionally, while robot actors possess a maximum of two capabilities, operators (denoted with green nodes) may even have four capabilities simultaneously.

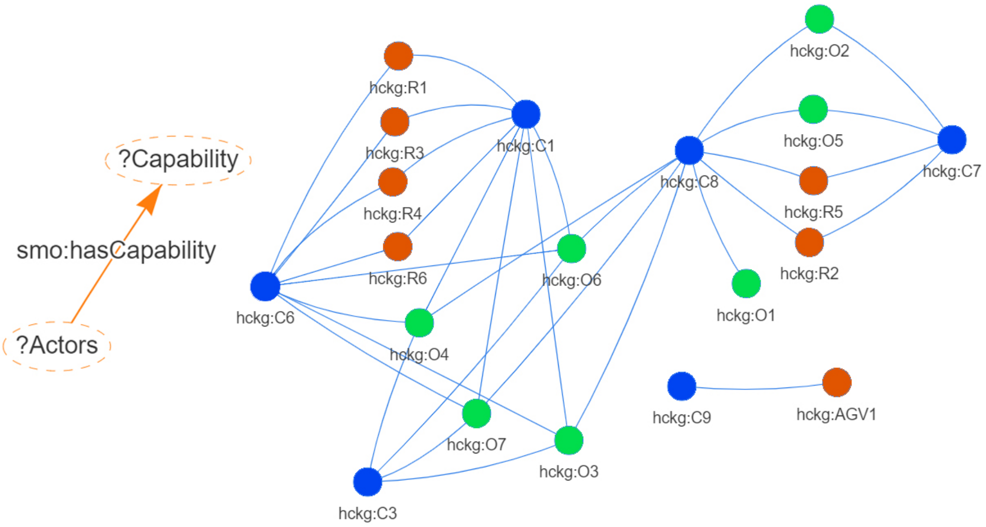
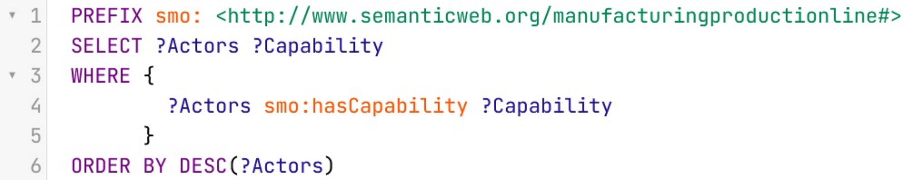


Figure I.3.14: Visualization of the *Actors* - *Capability* query (on the left-hand side) and the graph visualization of the result (on the right-hand side)



```
PREFIX smo: <http://www.semanticweb.org/manufacturingproductionline#>
SELECT ?Actors ?Capability
WHERE {
        ?Actors smo:hasCapability ?Capability
      }
ORDER BY DESC(?Actors)
```

Figure I.3.15: SPARQL query - Capability

A more complex data query is summarized in Figure I.3.16, to identify warning messages from sensors sent to DAS devices. First, the KG is reduced to the *sensor*, *observation* and *observed* nodes, which are also filtered down to the sensor individuals whose type names begin with "env" or "body", corresponding to environmental or body sensors. Next, further data are added to the list and characterised as *observationValue*, *warningLimit*, and *alarmLimit* of the affected data sets. Another filter is applied to identify the cases when the *observationValue* is higher than the *warningLimit*. Finally, the name of the DAS device, the message and the location of the equipment are listed. The detailed SPARQL query can be found in Figure I.3.17. On the right-hand side of Figure I.3.16, only the most relevant part of the query result is graphically visualized, where the purple node denotes the location of the sensor, the red ones represent the message sent to the DAS, and the green node corresponds to the specific operator to which the DAS device is equipped, e.g. smart glass. Regarding the graph in the bottom-right corner of the figure, the locations of the observing sensor and DAS device are identical as they are body sensors.

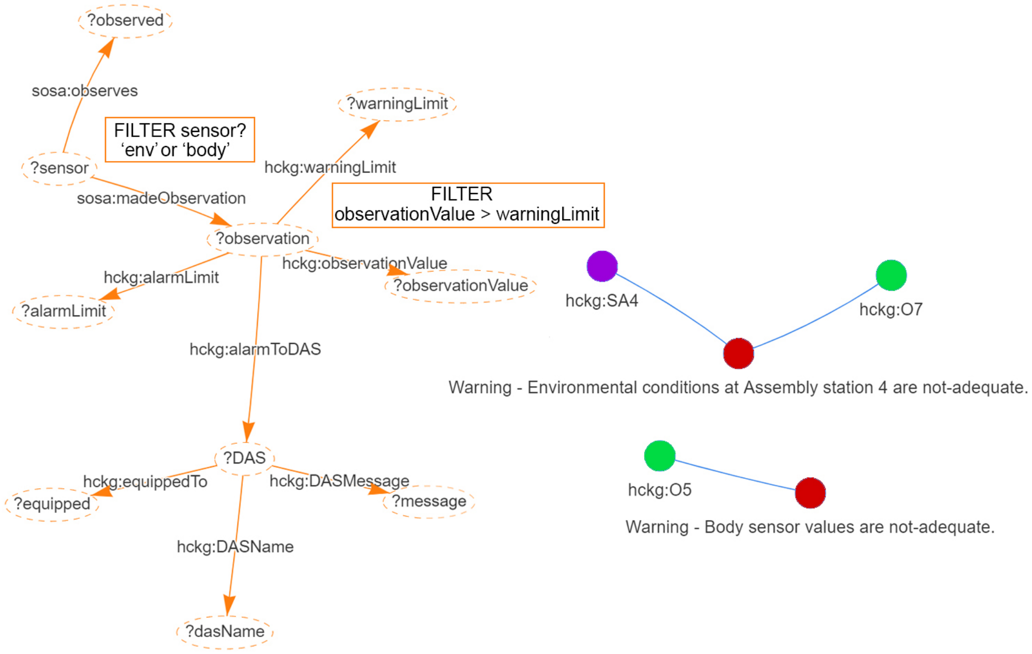


Figure I.3.16: Visualization of the *sensor* - *observation* - *DAS* query (on the left-hand side) and a graph visualization of the result (on the right-hand side)

```
PREFIX hckg: <https://www.abonyilab.com/wh-ontology#>
PREFIX sosa: <http://www.w3.org/ns/sosa#>
SELECT ?sensor ?observation ?observed ?observationValue ?warningLimit
?alarmLimit ?DAS ?message ?dasName ?equipped
WHERE {
        ?sensor sosa:madeObservation ?observation .
        ?sensor sosa:observes ?observed
        FILTER (regex(str(?sensor),"env") || regex(str(?sensor),"body"))
        ?observation  hckg:observationValue ?observationValue .
        ?observation hckg:warningLimit ?warningLimit .
        ?observation hckg:alarmLimit ?alarmLimit
        FILTER ( ?observationValue > ?warningLimit )
        ?observation hckg:alarmToDAS ?DAS .
        ?DAS hckg:DASMessage ?message    .
        ?DAS hckg:DASName ?dasName .
        ?DAS hckg:equippedTo ?equipped
    }
```

FIGURE I.3.17: SPARQL query - DAS

Figure I.3.18 represents the result of the SPARQL query in Figure I.3.19, which lists the *Result* individuals of specific *ManufacturingFacility*, namely $SA4$ and $SC4$, *Assembly station 4* and *Crimping station 4*, where *Operator 7* ($O7$), *Robot 6 and 7* ($R6 - R7$) are performing assembly activities, that create the visualized 15 different result (red nodes). Figure I.3.18 also serves as a visual analytic tool to investigate the collaboration between human-robot actors. The presented graph visualization methods can show supportive collaboration (type-4), such as $res63$ and $res66$ are performed at the same time on the same workpiece by $O7$ and $R6$. Another type-4 collaboration occurs in case of $res61$, performed by $O7$ and $R5$.

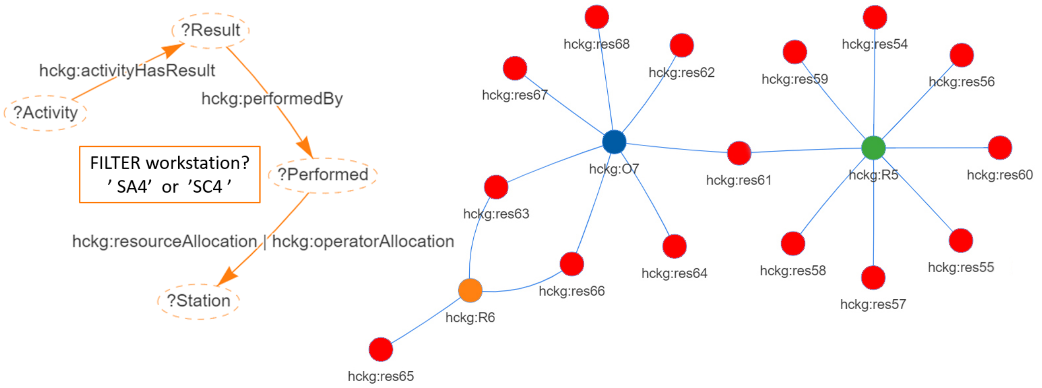
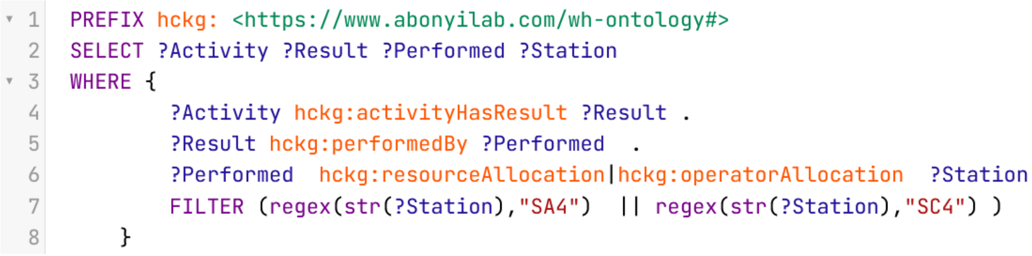


FIGURE I.3.18: Visualization of the *result - workstation - actor* query (on the left-hand side), and graph visualization of human-robot actors and the performed results at *Assembly station 4* and *Crimping station 4* (on the right-hand side)



```
PREFIX hckg: <https://www.abonyilab.com/wh-ontology#>
SELECT ?Activity ?Result ?Performed ?Station
WHERE {
        ?Activity hckg:activityHasResult ?Result .
        ?Result hckg:performedBy ?Performed  .
        ?Performed  hckg:resourceAllocation|hckg:operatorAllocation  ?Station
        FILTER (regex(str(?Station),"SA4")  || regex(str(?Station),"SC4") )
    }
```

FIGURE I.3.19: SPARQL query - result - workstation - actor

An application of the human-machine collaboration time KPI-related statement is given by the following result in Figure I.3.20, continuing the previous example with Operator 7 ($O7$) and *Robots 6-7* ($R6 - R7$), while working on *Assembly station 4*. The total times of supportive collaboration (type 4) are presented in the chart visualized in Figure I.3.18. It should be noted that $O7$ spent more assembly time collaborating supportively with *Robots 5 and 6* than performing individual work. Additionally, in the last two columns of the graph, simultaneous collaboration (type 3) is also highlighted, performed by *Operator 7* and *Robot 6*.

To visualize and analyse type 3 as well as the simultaneous collaborative assembly activity sequence, the results and precedence of the activities need to be investigated. Therefore, in Figure I.3.21, the result of a query on a KG is presented and visualized with directed graphs, creating precedence graphs. The detailed query in SPARQL language can be found in Figure I.3.22. The yellow nodes represent the activities, while the purple ones depict the results. The directed edges represent different object properties of the KG, namely:

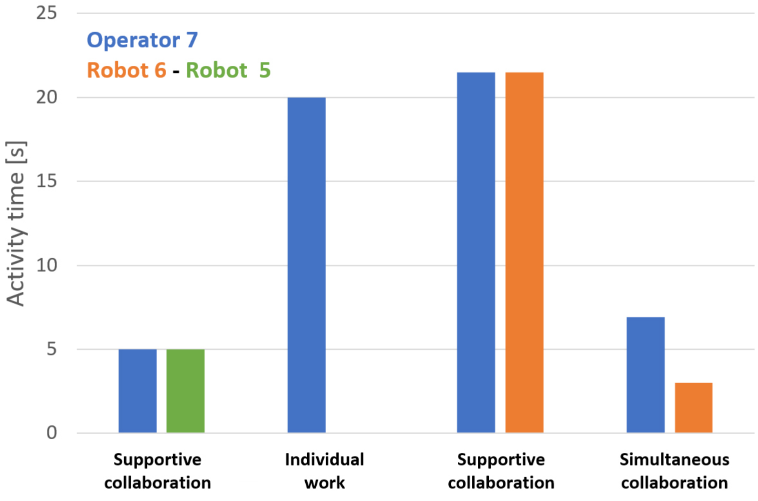


FIGURE I.3.20: Distribution of assembly work in terms of operator $O7$, including the total supportive, simultaneous and individual times

- **done** - *activityHasResult* object property
  Shows the result condition of a specific activity if the assembly task is accomplished.

- **prec.** - *activityHasPrecedence* object property
  Represents the precedence criteria of an activity that has to be carried out before the specific activity can be started.

- **perform** - *performedBy* object property
  Describes the human or robot actor that performs the activity.

The activities and results, which can serve as a basis for the analysis of process flow, where the sequence of procedures and criteria can be followed from activities $a75$ to $a83$ are shown on the left-hand side of Figure I.3.21. An extended visualization, where the *perform* edges are added showing that a human or robot actor has performed a specific activity, is presented on the right-hand side of this figure.

The study of the in- and out-degrees of a directed graph [209] makes it possible to create clusters [210] in the network. Utilizing this method, it can be stated that if a result node contains more than one *done* in-degree, it has been performed by type-4 supportive collaboration of actors, as it is labeled in the cases of activities $a77-a78$ and $a81-a82$. In these cases, the actors need to wait for the same result (precedence is given) before starting to perform different activities simultaneously on the same workpiece.

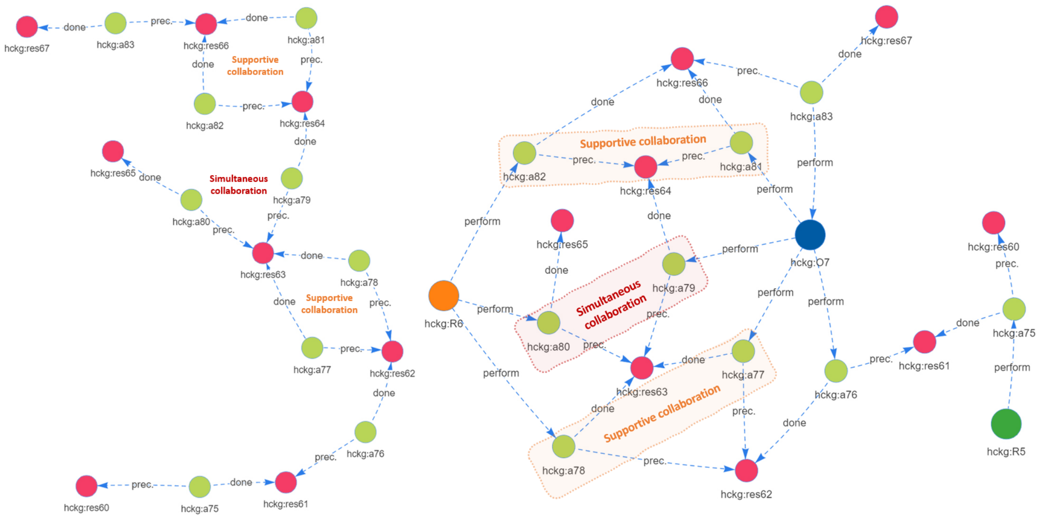
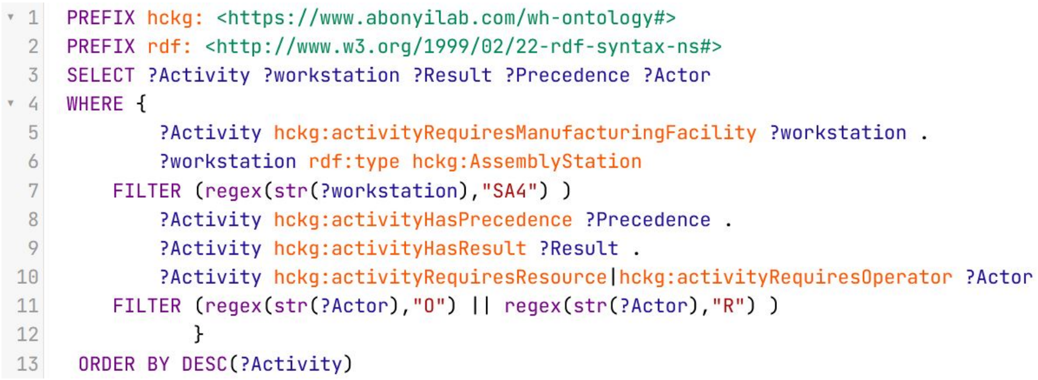


Figure I.3.21: Directed graph result and activity nodes (on the left-hand side) as well as the same result, including the human-machine actor nodes (on the right-hand side)



```
PREFIX hckg: <https://www.abonyilab.com/wh-ontology#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?Activity ?workstation ?Result ?Precedence ?Actor
WHERE {
        ?Activity hckg:activityRequiresManufacturingFacility ?workstation .
        ?workstation rdf:type hckg:AssemblyStation
    FILTER (regex(str(?workstation),"SA4") )
        ?Activity hckg:activityHasPrecedence ?Precedence .
        ?Activity hckg:activityHasResult ?Result .
        ?Activity hckg:activityRequiresResource|hckg:activityRequiresOperator ?Actor
    FILTER (regex(str(?Actor),"O") || regex(str(?Actor),"R") )
           }
 ORDER BY DESC(?Activity)
```

FIGURE I.3.22: SPARQL query - collaboration at station $SA4$

According to the precedence graph, if two (or more) activity nodes are given the same precedence (*prec.* edge) but yield different results (*done* edge), a type-3 simultaneous collaboration has occurred. It can also be observed in Figure I.3.21 that activities $a79$ and $a80$ are performed at the same time after being given the same precedence ($res63$), but yielding different results once completed ($res64$ and $res65$).

The final result in this section is a conceptual dashboard shown in Figure I.3.23, where the percentages represent the levels of competence of the operators and the conditions of the robots. The previously presented query result as well as the KPIs in Subsection I.3.3 can be a source of data for smart glass, dashboards on the shop floor, the DAS or other smart devices.

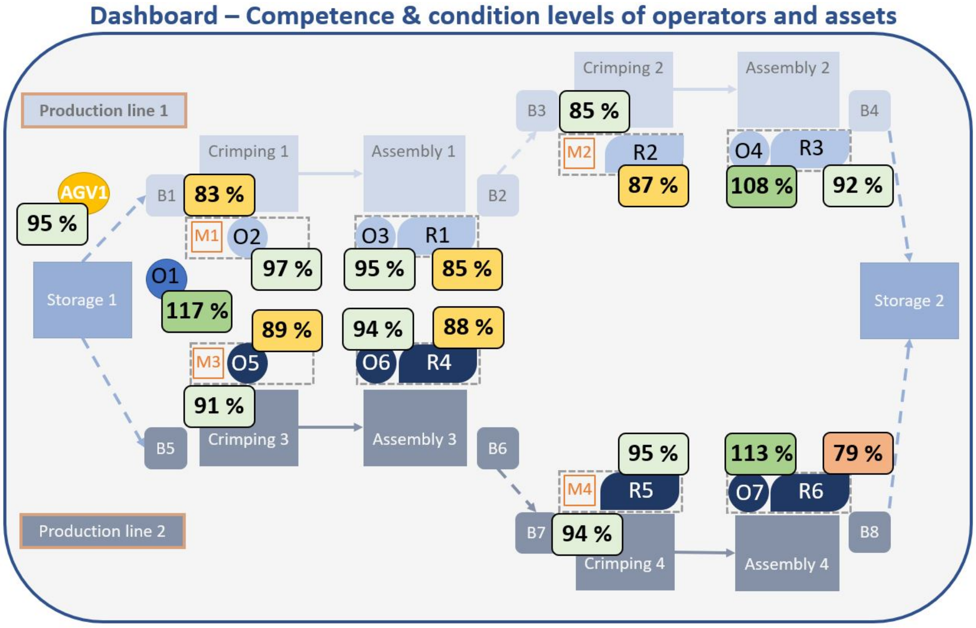


FIGURE I.3.23: Conceptual dashboard for human-centric manufacturing

In this section, the formation of the KG in this industrial case study and the analytical methods were presented with detailed examples. The following section summarizes the contributions of this chapter.

I.3.7 Summary of human-centric knowledge graph framework

The design concept of a human-centered knowledge graph (HCKG) based on industry standards and semantic technologies associated with Industry 5.0 technologies is presented in this chapter. An extended version of the MOM model and the development framework were introduced in the form of a layered structure. The activities performed by an operator full within the scope of this study, including the evaluation of movements, collaboration with machines, work steps and ergonomics amongst conditions. Additionally, it is highlighted that activity recognition technologies can enhance the utilizable data in a knowledge graph in a smart factory environment.

This chapter aimed to summarize the existing methods and tools of semantic development as well as proposed a concept to create standard models of human-centered collaboration, which has been demonstrated in an industrial use case. The contributions of this chapter are as follow:

- Highlighted the need for integrating human factors in cyber-physical systems.
- Suggested an extension of the automation standards (ISA-95, AutomationML, B2MML) with human-related processes and presented applications of semantic technologies.
- The concept was tested on a reproducible industrial case study. Several graph-based analyses were performed using normal, directed or hypergraphs such as resource allocation analysis, KPI evaluation and the integration of a DAS.
- The HCKG-based application made it possible to detect different types of collaboration between human and machine actors in the assembly process.
- Additionally, a conceptual application was proposed for a human-centric manufacturing dashboard.

Part II.

Network science based process optimization - Advanced manufacturing analytics

This part discusses the network science-based process optimization and presents several detailed application aspects in Chapters II.2-II.4. First, Chapter II.1 highlights the problem statement and introduces the theoretical and research background of network science-based process optimization, such as assembly line balancing, community detection or hypergraph-based analytics. Chapter II.2 presents a detailed method for solving assembly line balancing with the combination of analytic hierarchy process and multilayer network-based modeling. Additionally, a complex, multilayer analysis of a wire-harness assembly graph network is described. Chapter II.3 presents an efficient network community detection algorithm based on crossing minimization and bottom-up segmentation. Finally, Chapter II.4 discusses the hypergraph-based analysis of a collaborative manufacturing process.

Chapter II.1

Problem statement of network science-based process optimization

The previous part of this doctoral thesis showed a variety of applications and highlighted the advantages of semantic technologies in modern industry. Following the advised graph-based data access approach, this chapter aims to give an overview of some of the possible analytic methods and optimization procedures that can be utilized on graph networks. The motivation is to provide effective optimization for complex production processes of an Industry 4.0 environment and to handle the dynamically changing conditions and requirements on a shop floor.

The contents of this introduction section are the followings:

- First Section II.1.1 gives a summary about application options of semantic features for optimization.
- Section II.1.2 describes how to convert raw or ontology data into a graph network and create multilayer network representation.
- In Section II.1.3 the assembly line balancing problem is presented in general.
- Section II.1.4 discusses the field of community detection algorithms and methods.
- Finally, Section II.1.5 presents the hypergraph-based production analytics.

II.1.1 Application of semantic features for optimization

First, some of the main features of utilizing semantic technologies and graph analytics from a human-centered approach are presented in Table II.1.1 [211]. These analytical methods can help to monitor and understand HRE [212] as well as KPI [213] factors better. Additionally, an example of its application is given in Table II.1.1 for each network metric.

TABLE II.1.1: Knowledge graph metrics and analytical features

Network metrics	**Analytical features of KGs**
Centrality computation	Which are the critical objects in the network?
	Detect the most significant influencing factors in the operator's environment.
Similarities between nodes and edges	How similar are two objects based on their properties and how are they connected to other objects?
	Solve allocation problems concerning operators and resources.
Flows and paths	What is the shortest, cheapest or quickest way to perform a process step?
	Optimize the shop floor layout to best match operator needs.
Cycles	Are there any cycles in the graph? If so, where are they?
	Analyse tasks allocated to humans and machines in a collaborative work environment
Network communities	What communities can be found in the production network?
	Facilitate the design of human-machine collaboration or cell formation.

To discuss data interoperability briefly in the case of optimizing semantic data in other graph-based ways, the following section introduces how to convert product information into graph databases.

II.1.2 Convert data into graph network and multilayer network representation

Figure II.1.1 represents the three stages of data processing: converting raw data to a graph network and then to a multilayer graph network. The first step is to transform the collected data from the production system into ontology-based datasets (linked data as RDF). Then if the ontology skeleton is complete (also referred to as system modelling), connect the production datasets with the ontology. During the third phase, the created RDF based semantic network is turned into a multilayer network. To perform these tasks, vector- and matrix-sorted data aggregation methods can be utilized.

Usually, production systems include multiple subsystems and layers of connectivity. Thus, although research-based solutions for classical operations typically use a graph-based representation of problems and flow-based optimization algorithms, conventional single-layer networks quickly become incapable of representing the complexity and connectivity of all the details of the production line. With the overlapping data in Industry 4.0 solutions, it should be highlighted that multilayer networks are expected to be the most suitable options for representing modern production lines. The concept of a multilayer network was developed to represent multiple types of relationships [214], and these models have been proven to be applicable to the representation of complex connected systems [215]. Network-based models can also represent how products, resources and operators are connected [216], which is beneficial in terms of solving manufacturing cell formation problems [217].

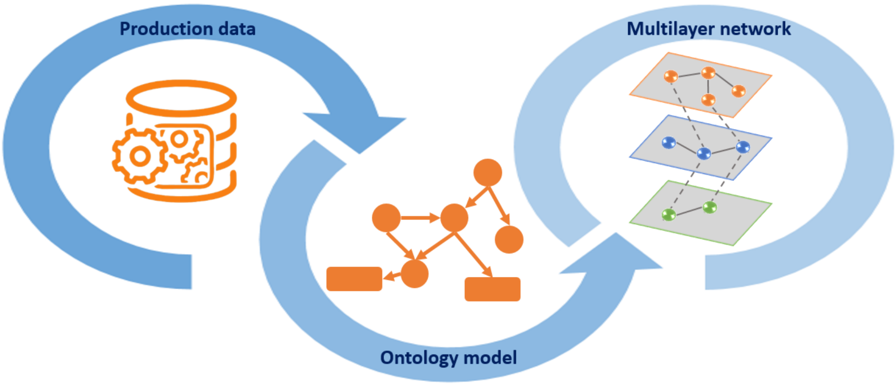


Figure II.1.1: The steps of data transformation towards creating a process-specific ontology and graph-based multilayer network

Once the data is accessible for optimization algorithms, different methods can be utilized, such as assembly line balancing or community detection, as presented in the following two sections.

II.1.3 Algorithmic solutions to the assembly line balancing problem

This section briefly presents the assembly line balancing problem in production, which is one of the most common optimization tasks.

Assembly lines in production are still one of the most widely applied manufacturing systems [218]. Assembly-Line Balancing (ALB) [219] deals with the balanced assignment of tasks to the workstations, resulting in the optimization of a given objective function without violating precedence constraints [220]. The efficiency of these optimization tasks is mostly determined by the model of the manufacturing process represented [221].

Line balancing is a non-deterministic polynomial-time hard (NP-hard) optimization problem, which means that the computational complexity of the optimization problem increases exponentially as the dimensions of the problem increase. This challenge explains why numerous approaches such as simulated annealing (SA) [222, 223], hybrid heuristic optimization [223, 224], chance-constrained integer programming [225], recursive and dynamic programming [226], as well as tabu search [227] have been utilized in the field of production management. The fuzzy set theory provides a transparent and interpretable framework to represent information uncertainty and solve the ALB problem [228, 229]. Among the wide range of heuristic methods capable of achieving reasonable solutions [230], SA is the most widely used search algorithm [231], so it has already been applied to solve mixed and multi-model line-balancing problems [232].

The following section continues with another common optimization problem of complex networks, with community detection.

II.1.4 Community detection algorithms

The community structure is one of the essential features of networks [233, 234]. Community detection algorithms are fundamental tools to uncover how the networks are structured [235]. Identifying communities and their boundaries are crucial to classifying nodes according to their structural position in the community [236, 237]. Recently, more and more community detection algorithms are appearing [238, 239, 240, 241]. In the case of large networks [242, 243], the modularity optimization-based approach is highly studied and applied [244, 245] because of the outstanding effectiveness and low computation time.

Many community detection algorithms are based on node sorting, and serialization, where the basic idea is to use an efficient clustering in the order of network nodes [246, 247, 248]. Serialization is the basis of community detection procedures in networks, which is directly related to modularity property and modularity optimization algorithms as spectral clustering [249, 250]. The spectral clustering method consists of transforming the initial set of objects into a set of points in space, whose coordinates are elements of eigenvectors: the set of points is clustered via standard techniques [248, 251].

After the general optimization method, the following section presents a more complex analytic method with the so-called hypergraphs.

II.1.5 Introduction to hypergraph-based analytics

A hypergraph is a mathematical structure consisting of a set of nodes and a set of hyperedges, where each hyperedge is a subset of nodes. Unlike in a traditional graph where edges connect pairs of nodes, hyperedges in a hypergraph can connect any number of nodes [252].

Hypergraphs provide a sufficient description of a system with hierarchical and multilevel model techniques to describe collaboration between larger groups or complex networks. In operations research, one of the most dynamically developing fields is related to hypergraphs [253, 252] and higher order interactions [254]. In traditional networks, only pairwise interactions are defined within the vertices, which is suitable for describing collaborations between two participants but insufficient in the case of complex networks, describing collaborations between larger groups. Therefore, formalising a multilayer higher-order network can help uncover network properties such as the community structure, various centrality measures [254] and efficient clustering of data [255]. Hypergraphs are also increasingly being used in cooperative game theory [256, 257] as well as in cooperative multi-agent reinforcement learning [258].

It is believed that hypergraph models will be much more widely applied in the analysis and design of manufacturing systems. The applicability of this modeling approach has already been proven in the design of knowledge-centric robot systems where, based on a structural meta-model and the related domain-specific language, a hypergraph has been designed [259].

The analytical techniques of hypergraphs can also support the design of smart manufacturing applications. The clustering-based Cloud Manufacturing Service Management Model has been developed to manage the high number of instances in which dynamically changing cloud services are applied, using three different layers [260]. Allocation problems of flexible manufacturing systems, e.g. tool switching problems, can also be handled more efficiently with hypergraphs [261]. Another application is a hypergraph convolutional network, developed to predict the removal rate of material in chemical mechanical planarization, the benefit of which is to identify the structure of underlying equipment containing essential interaction mechanisms among different components [262].

As a manufacturing cell can be identified as a part of a hypergraph, it can also support the field of cell formation in Flexible Manufacturing, creating multidimensional layout diagrams and analysing the internal mechanism [263]. Furthermore, the framework of a hypergraph can facilitate the optimal model-based decomposition (OMBD) of engineering design problems [264]. A Cell Formation algorithm called Hypergraph BFS (Breadth-First Search) has been developed by Kandiller, which is another efficient machine-grouping procedure [265]. The algorithm examines the vertex set of the hypergraph and tries to form machine cells based on their similarities by partitioning the dataset as well as selecting key vertices and using them as roots in each search process [266]. Furthermore, hypergraphs can also support allocation problems in the era of Industry 4.0, e.g. in robot task allocation, where a multilevel framework is required to handle assignments [267, 268].

Another field where the outstanding network analysis techniques of hypergraphs can be utilized is competency mapping, an approach highlighting expertise, reusing knowledge or monitoring key performance indicators (KPI) to increase productivity. The investigation of multi-hypergraph structures offers an effective solution for competency mapping [269].

Chapter II.2

Analytic hierarchy process and multilayer network-based method for assembly line balancing

This Chapter introduces an optimization method that combines the analytic hierarchy process and the multilayer network-based production representation to solve the assembly line balancing problem.

Assembly line balancing improves the efficiency of production systems by the optimal assignment of tasks to operators. The optimization of this assignment requires models that provide information about the activity times, constraints and costs of the assignments. A multilayer network-based representation of the assembly line-balancing problem is proposed, in which the layers of the network represent the skills of the operators, the tools required for their activities and the precedence constraints of their activities. The activity-operator network layer is designed by a multi-objective optimization algorithm in which the training and equipment costs as well as the precedence of the activities are also taken into account. As these costs are difficult to evaluate, the analytic hierarchy process (AHP) technique is used to quantify the importance of the criteria. The optimization problem is solved by a multi-level SA algorithm, that efficiently handles the precedence constraints. The efficiency of the method is demonstrated by a case study from wire harness manufacturing (described in the Appendix C.1), which is a subset of the complex wire harness assembly production model, discussed earlier in Chapter I.2.

In the proposed novel network model, the layers represent the skills of the operators, the tools required for the activities, and the precedence constraints of the activities. At the same time, a multi-objective optimization algorithm designs the assignment of activities and operators to network layers. The proposed multilayer network approach supports the intuitive formulation of multi-objective line balancing optimization tasks. Besides the utilization of operators, the utilization of the tools and the number of skills of an operator are also taken into account. The main advantage of the proposed network-based representation is that the latter two objectives are directly related to the structural properties of the optimized network.

To deal with the complexity of the ALB problem, an SA algorithm was also developed. The developed algorithm utilizes a unique problem-oriented sequential representation of the assignment problem and applies a neighbourhood-search strategy that generates feasible task sequences for every iteration. Since the algorithm has to handle multiple aspects of line balancing, the AHP technique is used to quantify the importance of the objectives, also known as Saaty's method [270]. AHP is a method for multi-criteria decision-making which is used to evaluate complex multiple criteria alternatives involving subjective judgments [271]. This method is a useful and practical approach to solving complex and unstructured decision-making problems by calculating the relative importance of the criteria based on the pairwise comparison of different alternatives [272]. The method has been widely applied thanks to its effectiveness and interpretability. Two papers were found in which it has already been applied to determine the cost function of multi-objective SA optimization problems. In the first case study of supplier selection, AHP was applied to calculate the weight of every objective by applying the Taguchi method [273] (Adaptive Tabu Search Algorithm - ATSA [274]). In contrast, in the second report, this concept was applied to the maintenance of road infrastructure [275].

The novelties of this chapter are the following:

- In Section II.2.1, the main problem formulation is introduced, including the multilayer network representation of multiple aspects concerning the balancing of production lines, and the details of the objective function.

- In Section II.2.2, an SA algorithm is introduced based on a novel sequential representation of the line-balancing problem and the search algorithm that guarantees the fulfilment of the precedence constraints.
- Section II.2.3 demonstrates how AHP can be used to aggregate the multilayer network-represented objectives of the line-balancing problem for SA.
- Section II.2.4 presents the parameter testing result of the proposed method.
- In Section II.2.5 a more complex multilayer analysis is presented, which includes community detection as well.
- Finally, Section II.2.6 summarizes the contributions of this chapter.

II.2.1 Problem formulation of multilayer based, multi-objective assembly line balancing

In this section, the problem formulation is presented. First, the representation of production line modelling with the multilayer network is introduced in Section II.2.1.1. The details of the minimized function and its AHP-based aggregation are given in Section II.2.1.2.

II.2.1.1 Multilayer network-based representation of production lines

The proposed network model of the production line consists of a set of bipartite graphs that represent connections between operators, $\mathbf{o} = \{o_1, \ldots, o_{N_o}\}$; skills of the operators needed to perform the given activity, $\mathbf{s} = \{s_1, \ldots, s_{N_s}\}$; equipment, $\mathbf{e} = \{e_1, \ldots, e_{N_e}\}$; activities (operations), $\mathbf{a} = \{a_1, \ldots, a_{N_a}\}$; and the precedence constraints between activities, $\mathbf{a}^{'} = \left\{{a_1}^{'}, \ldots, a_{N_a}^{'}\right\}$. The relationships between these sets are defined by bipartite graphs $G_{i,j} = (O_i, O_j, E_{i,j})$ represented by $\mathbf{A}[O_i, O_j]$ biadjacency matrices, where O_i and O_j denote a general representation of the sets of objects, such that $O_i, O_j \in \left\{\mathbf{s}, \mathbf{e}, \mathbf{a}^{'}, \mathbf{a}, \mathbf{o}\right\}$.

The edges of these bipartite networks represent structural relationships; e.g., the biadjacency matrix $\mathbf{A}[\mathbf{a}, \mathbf{a}']$ represents the precedence constraints or $\mathbf{A}[\mathbf{a}, \mathbf{o}]$ represents the assignments of activities to operators. Moreover, the edge weights can be proportional to the number of shared components/resources or time/cost (see Table II.2.1) [216].

Table II.2.1: Definition of the biadjacency matrices of the bipartite networks used to illustrate, how a multidimensional network can represent a production line.

	Nodes	**Description**
$\overline{\mathbf{W}}$	Activity ($\mathbf{a}$) - Operator ($\mathbf{o}$)	Operator assigned to the activity
$\mathbf{S}$	Activity ($\mathbf{a}$) - Skill ($\mathbf{s}$)	Skill/education required for a category of activities
$\mathbf{E}$	Activity ($\mathbf{a}$) - Equipment ($\mathbf{e}$)	Equipment which is in use in an activity
$\mathbf{A}'$	Activity ($\mathbf{a}$) - Activity ($\mathbf{a}'$)	Precedence constraint between activities

As can be seen in Figure II.2.1, these bipartite networks are strongly connected. The proposed model can be considered as an interacting or interconnected network [214], where bipartite networks define the layers. Since different types of connections are defined, the model can also be handled as a multidimensional network. As illustrated in Figure II.2.2, when relationships between the sets O_i and O_j are not directly defined, it is possible to evaluate the relationship between their elements $o_{i,k}$ and $o_{j,l}$ in terms of the number of possible paths or the length of the shortest path between these nodes [216].

In the case of connected unweighted multipartite graphs, the number of paths intersecting the set O_0 can be easily calculated based on the connected pairs of bipartite graphs as follows:

$$\mathbf{A}_{O_0}[O_i, O_j] = \mathbf{A}[O_0, O_i]^T \times \mathbf{A}[O_0, O_j] \,. \tag{II.2.1}$$

In the proposed network model, the optimization problem is defined by the allocation of tasks that require the allocation of different skills and tools to an operator that might necessitate extra training, labour and investment costs. The main benefit of the proposed network representation is that these costs can be directly

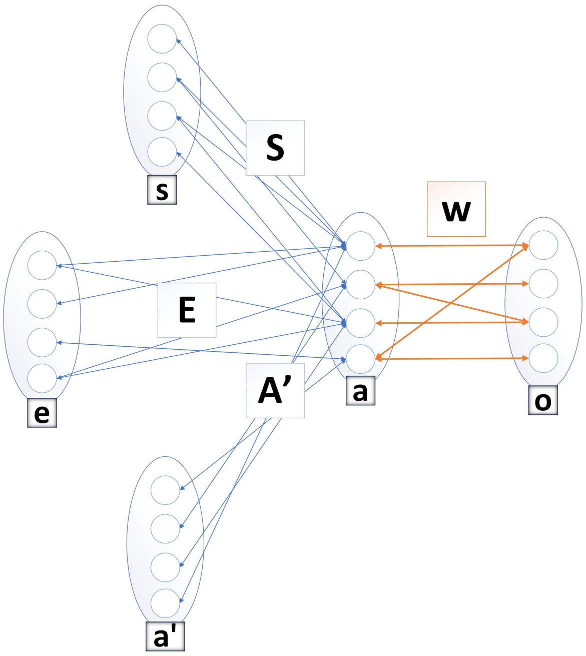


Figure II.2.1: Illustrative network representation of a production line. The definitions of the symbols are given in Table II.2.1.

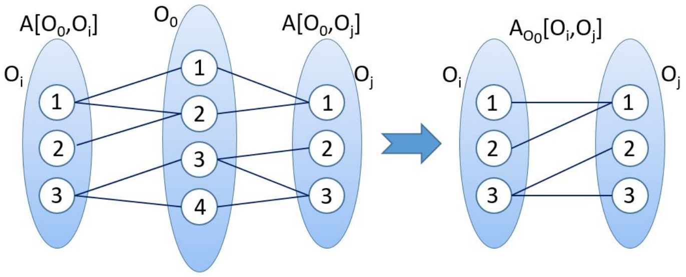


Figure II.2.2: Projection of a property connection

evaluated based on the products of the biadjacency matrices $\mathbf{S}$ and $\mathbf{W}$:

$$\mathbf{A}[\mathbf{s},\mathbf{o}] = \mathbf{A}[\mathbf{s},\mathbf{a}]\mathbf{A}[\mathbf{a},\mathbf{o}] = \mathbf{S}\mathbf{W}. \tag{II.2.2}$$

The resultant network $\mathbf{A}[\mathbf{s},\mathbf{o}]$ represents how many times a given skill should be utilized by an operator, while its unweighted version $\mathbf{A}^u[\mathbf{s},\mathbf{o}]$ models which skills the operators should have.

The design of the presented network model is based on the analysis of the semantically standardized models of production lines [276], and the experience gained in the development project connected to the proposed case study. The details of the multilayer network-based modelling of a wire-harness production process can be found in [216].

II.2.1.2 The objective function of assembly line balancing

A simple assembly line balancing problem (SALBP) assigns N_a tasks/activities to N_o workstations/operators. Each activity is assigned to exactly one operator, and the sum of task times of workstation should be less or equal to the cycle time T_c [277]. Precedence relations between activities must not be violated [278]. There are two significant variants of this problem [279], SALBP-1 aims to minimize N_o for a given T_c, while the goal of SALBP-2 is to minimize T_c for a predefined N_o [218, 280, 281]. In this work, the SALBP-2 problem was investigated and extended to include the following skill and equipment-related objective functions as described with equations in the followings.

Station-time-related objective: The main objective of line balancing is to minimize the cycle time T_c, which is equal to the sum of the maximum of the station times T_j. The utilization of the whole assembly line can be calculated as follows:

$$T_c = \arg\max_j T_j = \sum_{i=1}^{N_a} w_{i,j} t_i, \tag{II.2.3}$$

where t_i represents the elementary activity times of the a_i-th activity.

As the theoretical minimum of T_c is

$$T_c^* = \frac{\sum_{i=1}^{N_a} t_i}{N_o.}, \tag{II.2.4}$$

the following ratio evaluates the efficiency of the balancing of the activity times:

$$Q_T(\pi) = \frac{T_c^*}{T_c} = \frac{\frac{\sum_{i=1}^{N_a} t_i}{N_o}}{\sum_{i=1}^{N_a} w_{i,j} t_i.} \tag{II.2.5}$$

Skill-related (training) objective: The training cost is calculated with the node degree between skill-operator elements $s - o$. The number of skills N_s is divided by the sum of the node degrees k_i between sub-networks s and o in the multilayer representation:

$$Q_S(\pi) = \frac{N_s}{\sum_i^{s-o,o} k_i.} \tag{II.2.6}$$

Equipment-related objective function: The equipment cost is calculated with the node degree between equipment-operator elements $e - o$. The number of pieces of equipment N_e is divided by the sum of the node degrees k_i between sub-networks e and o in the multilayer representation:

$$Q_E(\pi) = \frac{N_e}{\sum_i^{e-o,o} k_i.} \tag{II.2.7}$$

Since the importance of these objectives is difficult to quantify, a pairwise comparison is used to evaluate their relative importance, and the AHP is used to determine the weights λ in the objective function:

$$Q(\pi) = \lambda_1 Q_T(\pi) + \lambda_2 Q_S(\pi) + \lambda_3 Q_E(\pi), \tag{II.2.8}$$

where $Q_T(\pi) \in [0, 1]$ represents the balance of the production line, and $Q_S(\pi) \in [0, 1]$ and $Q_E(\pi) \in [0, 1]$ measure the efficiency of how the skills and tools are utilized, respectively.

The application of AHP-based weighting is beneficial to integrate the normalised values of the easy to evaluate station-time and equipment-related objectives, and the less specific training-related costs. Although the pairwise comparison of the importance of these objectives and cost-items is subjective, the consistency of the comparisons can be evaluated based on the numerical analysis of the resulted comparison matrices (which will be shown in the next section), which clarifies the reason for the choice of AHP as an ideal tool to extract expert knowledge for the formalisation of the cost function.

II.2.2 Simulated annealing-based line-balancing optimization

This section presents the proposed optimization algorithm, introduces the representation of the SA problem, and discusses how the precedence constraints of the activities are represented. Additionally, it is presented how a sequencing problem formulates the assignment of activities to operators that the proposed simulated annealing algorithm can efficiently solve.

In the proposed network representation (Figure II.2.1), the assignment of activities to operators is defined by the elements $w_{i,j}$ of the matrix $\mathbf{W}$ that represent the i^{th} activity assigned to the j^{th} operator. Instead of the direct optimization of these $N_a \times N_o$ elements, a sequence $N_\pi = N_a + N_o - 1$ is optimized, where N_a represents the number of activities and N_o denotes the number of operators.

The concept of sequence-based allocation is illustrated in Figure II.2.3, where the horizontal axis represents the fixed order of the operators o_j and the vertical axis stands for the activities a_i , where $\pi(i)$ represents the index of the activity by the i^{th} sequence number. The ordered activities are assigned to the operators by $N_o - 1$ boundary elements, represented as $a_{\pi(i)} = *$, which ensure that the next activity in the sequence is assigned to the following operator.

In addition to these three objectives of the simulated production line, a so-called soft limit is also defined, which is the amount of the unaccomplished precedence of the activities (A'). This limitation of the order with regard to the activities is stored in the multilayer network.

The completion of a task is a precondition for the start of another because tasks depend on other tasks. The π sequence has some constraining condition and cannot

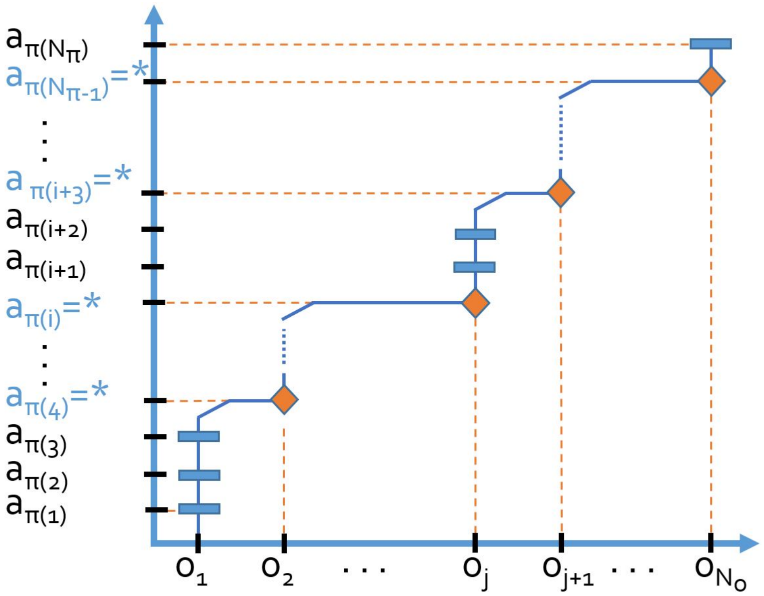


FIGURE II.2.3: Illustration of the sequencing method. The activities are separated into the different groups of activities that are assigned to different operators

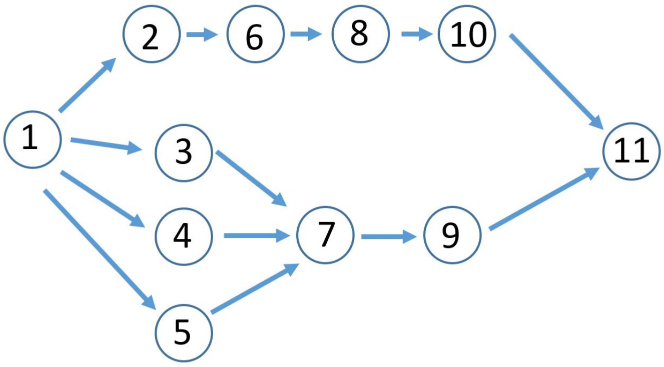


Figure II.2.4: Precedence graph of the example problem taken from Jackson [284, 279]

be entirely arbitrary. The precedence graph is used to represent these dependencies in SALBP [218, 282, 283]. Figure II.2.4 shows a problem from a well-known example by Jackson [284] with $N_a = 11$ tasks, where task 7 requires tasks 3–5 to be completed directly (direct predecessor) and task 1 indirectly (indirect predecessor). The precedence graph can be described by matrix $A'(i,j)$, $i, j = 1, 2, \ldots, N_a$, where $A'(i,j) = 1$ if task i is the direct predecessor of task j, otherwise, it is 0 [279]. The precedence graph is partially ordered if tasks cannot be performed in parallel. It must be determined whether a permutation $\pi = (\pi_1, \pi_2, \ldots, \pi_{N_a})$ is feasible or not according to the precedence constraint.

Based on the transitive closure A^* of A', π is feasible if $A^*(p_j, p_i) = 0, \forall i, j, i < j$; otherwise, π is infeasible [279]. A sub-sequence $(\pi_i, \pi_{i+1}, \ldots, \pi_j)$, where $i < j$ of π, can be defined by $\pi_{(i:j)}$. For example, a feasible sequence π of the precedence graph in Figure II.2.4 is $\pi = (1, 4, 3, 2, 5, 7, 6, 8, 9, 10, 11)$ and $\pi_{(2:4)} = (4, 3, 2)$ is a sub-sequence of π.

As will be presented in the next subsection, the key idea of the algorithm is that it determines the interchangeable sets of activity pairs and uses these in the guided simulated annealing optimization.

The developed optimization algorithm is shown in Algorithm 1 and consists of the following steps, which are the main novelties with the boundaries integration:

- Generating the initial feasible sequence.
- SA I: Optimization of the sequences of the activities.
  - SA II (embedded in SA I): in the case of a specific sequence, the activities are assigned to the operators by optimizing the location of the

boundary elements in sequence π as has been presented in Figure II.2.3, so SA I uses a cost function that relates to the optimal assignment.

Thank to the integration of the embedded two SA algorithms with the π boundary elements the method can result a more fast and favorable sequence.

Algorithm 1 Pseudocode of the proposed SA-ALB algorithm

```
Input: $s, e, Time, Precedence$
Output: $\pi, Q(\pi)$
Annealing: $maxiter, T, T_{max}, T_{min}, m_{max}, m_{min}$
$\alpha = (\frac{T_{min}}{T_{max}})^{1/maxiter}, T^1 = T_{max}$
 $\alpha_m = (\frac{m_{min}}{m_{max}})^{1/maxiter}, m^1 = m_{max}$

Begin
 $T = T_{max}$; $T_{max}$ = maximum value of temperature
 while $T < T_{min}$; $T_{min}$ = minimum value of temperature

 Generate initial sequence $\pi$, which satisfies the constraints
 Generate initial placement of the boundary elements
 Evaluate the cost function $Q(\pi)$, as functions (II.2.5), (II.2.6) and (II.2.7)

for $i = 1$ to $maxiter$ do
    Select randomly one interchangeable activity pair
        Interchange the activities and evaluate the new solution by implementing
        SA II that optimizes the placement of the boundary elements in this sequence
        // SA II is working with the same principle as this main SA I
    $NewQ(\pi_{new}) = Q(\pi_{new})$
     $\Delta = Q(\pi_{new}) - Q(\pi)$

    if $\Delta < 0$ then
        $\pi = \pi_{new}$
         $Q(\pi) = Q(\pi_{new})$

        else
            if $random() < exp(\frac{-\Delta}{T^i})$ then
                $\pi = \pi_{new}$
                 $Q(\pi) = Q(\pi_{new})$

    $T^{i+1} = \alpha T^i, m^{i+1} = \alpha_i m^i$
     End
```

II.2.3 Solving ALB with multilayer and AHP approach

The development of the proposed line-balancing algorithm is motivated by a development project which was defined to improve the efficiency of an industrial wire harness manufacturing process [285]. In this work, a subset of this model is used which consists of 24 activities, five operators, six skills and eight pieces of equipment as described in more details in Appendix C.1 (Chapter C). The applied benchmark data illustrates that the practical implementation of line balancing problems is also influenced by how much equipment is needed for the designed production line and how many skills should be learnt by the operators.

All the collected information is transformed into network layers, as shown in Figure II.2.5. The top of the figure shows the bipartite networks that represent the details of the assignments, while the bottom of the figure represents the tree layers of the network that define the activity–operator, skill–operator, and equipment–operator assignments. As can be seen, this representation is beneficial as it shows how similar operators, skills and equipment can be grouped into clusters.

Although this is not shown in the figure, the weights of the edges represent the costs or benefits of the assignments. The final form of the network is formed based on a multi-objective optimization of the sets of active edges.

As some of these objectives are difficult to measure, the proposed AHP-based method is utilized to convert the pairwise comparisons of the experts into weights of criteria. The structure of the decision problem is represented in Figure II.2.6. As this figure illustrates, the AHP is used to compare difficult to evaluate equipment and skill assignment costs and the importance of the objectives. The pairwise comparison was performed by a process engineer, and the resulting comparison matrices can be found in Tables II.2.2–II.2.4. Based on the analysis of the the eigenvalues of these matrices [272], it can be stated that the evaluations were consistent.

Since the activities cannot be performed in parallel, a precedence graph defines the most crucial question, namely whether a permutation of sequence π is feasible. Based on the transitive closure of the adjacency matrix of the graph, the interchangeable sets of activities can be defined as depicted in Figure II.2.7.

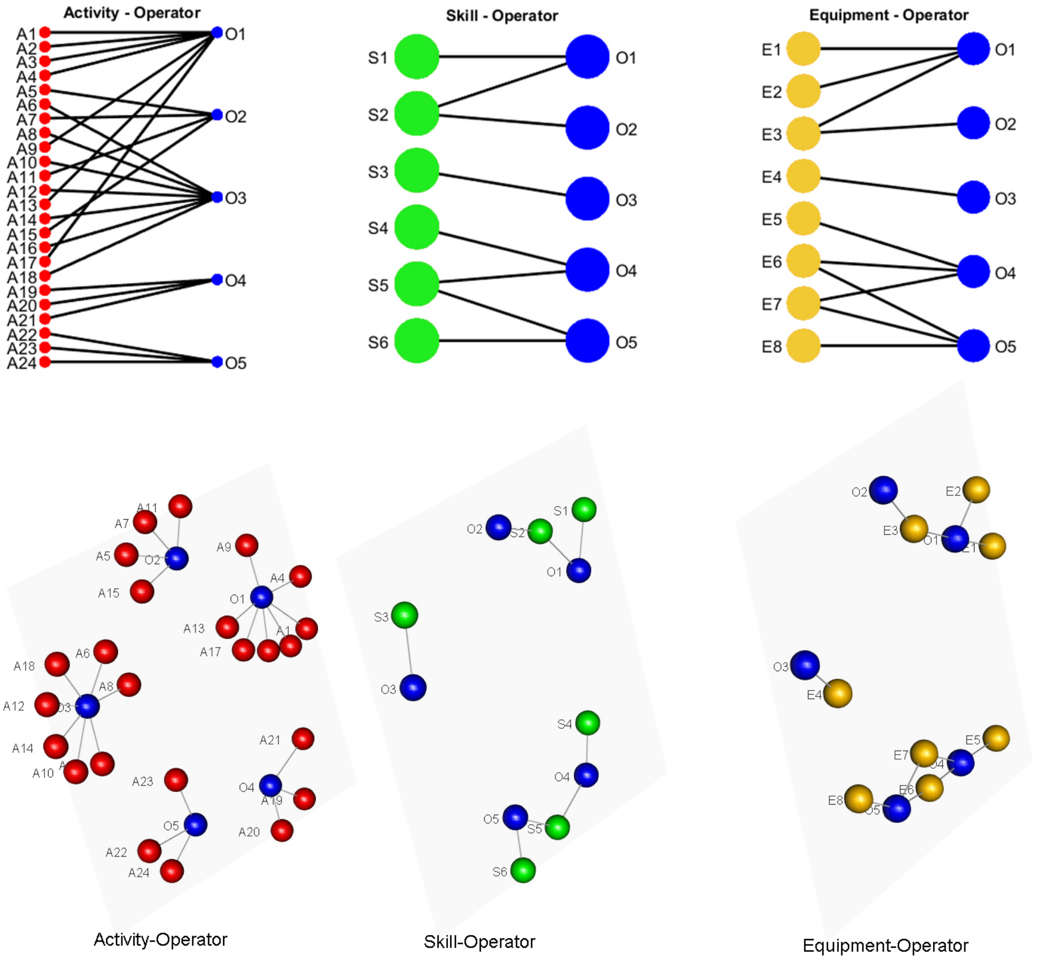


FIGURE II.2.5: Illustration of the skill-operator and equipment-operator assignments after line balancing

The result of the optimization is shown in Figure II.2.5, which illustrates that the five operators assigned to different skills and pieces of equipment.

The reliability and the robustness of the proposed method are evaluated by ten independent runs of the optimization algorithm to highlight how the stochastic nature of the proposed method influences the result, as well as showing the effect of the number of operators on the solutions. The aim of the analysis of the independent runs was to estimate the variance of the solutions caused by the stochastic nature of the process and the optimization algorithm. The sample size of such repeat studies can be determined based on the statistical tests of the estimated variance. In the analysis, ten experiments were found to get proper estimation of the variance (which is in line with the widely applied ten-fold cross-validation concept).

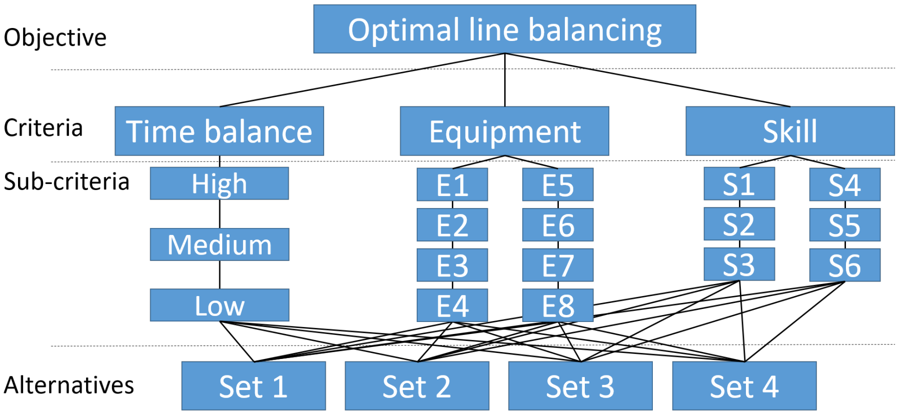


FIGURE II.2.6: Analytic hierarchy process (AHP) used to solve a decision problem

TABLE II.2.2: AHP TOP matrix that shows the relative importance of the objectives. It can be seen that, in this pair-wise comparison, the skill-related cost is evaluated as being twice as important as the equipment-related costs.

	Balancing	**Equipment**	**Skill**
Balancing		2.00	4.00
Equipment	0.50		2.00
Skill	0.25	0.50	

TABLE II.2.3: AHP equipment matrix that shows the relative importance of the equipment.

	E1	**E2**	**E3**	**E4**	**E5**	**E6**	**E7**	**E8**
E1		1	0.33	0.50	2	3	3	2
E2	1		0.33	0.50	2	3	3	2
E3	3	3		2	5	7	7	5
E4	2	2	0.50		3	5	5	3
E5	0.50	0.50	0.20	0.33		2	2	1
E6	0.33	0.33	0.14	0.20	0.50		1	0.50
E7	0.33	0.33	0.14	0.20	0.50	1		0.50
E8	0.50	0.50	0.20	0.33	1	2	2	

Table II.2.4: AHP skill matrix that shows the relative importance of the skills.

	S1	**S2**	**S3**	**S4**	**S5**	**S6**
S1		0.20	0.30	0.50	0.50	0.14
S2	5		2	3	3	0.50
S3	3	0.50		2	2	0.33
S4	2	0.30	0.50		1	0.20
S5	2	0.30	0.50	1		0.20
S6	7	2	3	5	5	

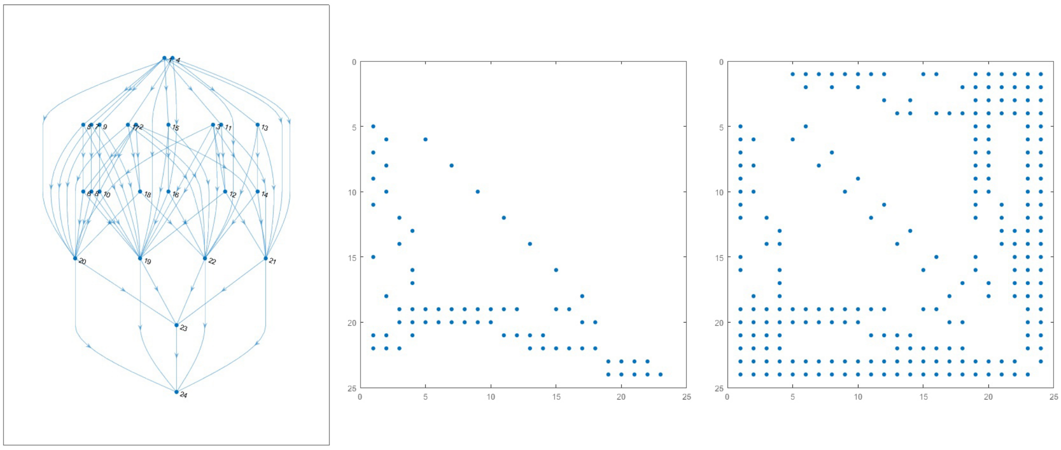


Figure II.2.7: Possible path (left), precedence (middle) and transitive closure (right) of the activities (the unmarked pairs are interchangeable)

Figure II.2.8 shows the different total activity times of each operator during the simulation. In this case, the station times do not differ greatly, and the result is optimal [216]. The developed algorithm was implemented in MATLAB and the presented problem can serve as a benchmark for constrained multi-objective line balancing. Additionally, a red horizontal line in the figure shows the $\overline{T_c}$ mean cycle time of the five operators.

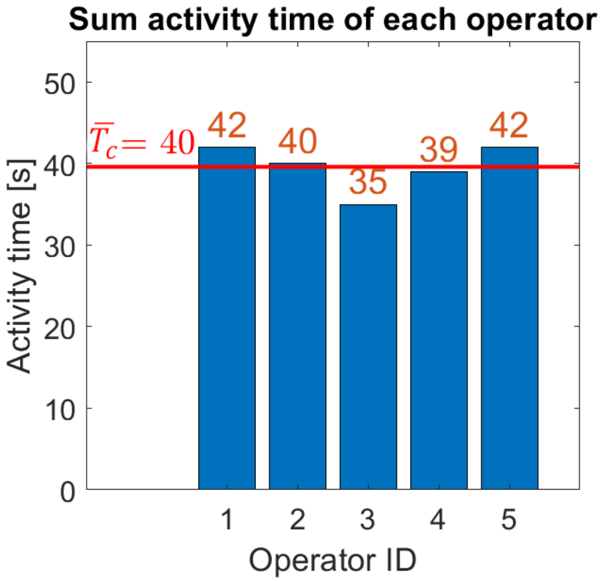


FIGURE II.2.8: Summarized activity time comparison of the five allocated operators

II.2.4 Parameter testing

This section presents the parameter testing results of the assembly line balancing problem, solved with proposed method.

In the first test, the modification of the cost function has been tested, as apply all three cost elements, compared to use only the time-related goal. Figure II.2.9 represents two cases, which have the biggest difference from each other in terms of the objective function. The top of the figure shows the original case of the assembly line balancing, where all three types of cost are taken into account, such as time, skill, and equipment-related costs, and the algorithm uses the AHP to handle their hierarchy. Additionally, with all three objective function elements, the costs are the followings: 94% of time cost, 38% of skill cost and 36% of equipment cost. The second case (at the bottom of the figure) represents a simplified cost function, where only the time-related optimization goal is applied. The simplification of the objective function results an even better, 97% of time cost. As the skill and equipment objectives were not taken into account, the skill-operator and equipment-operator allocations are also different, and the edge numbers are increased. Compared to the first scenario, the number of skill-operator allocations (edges) changed from 8 to 12, furthermore, the number of equipment-operator allocations also changed from 11 to 15. In conclusion, it can be stated, that the modification of the cost function had a significant influence on the number of skill and equipment assignments to operators, while resulting slightly better cycle times.

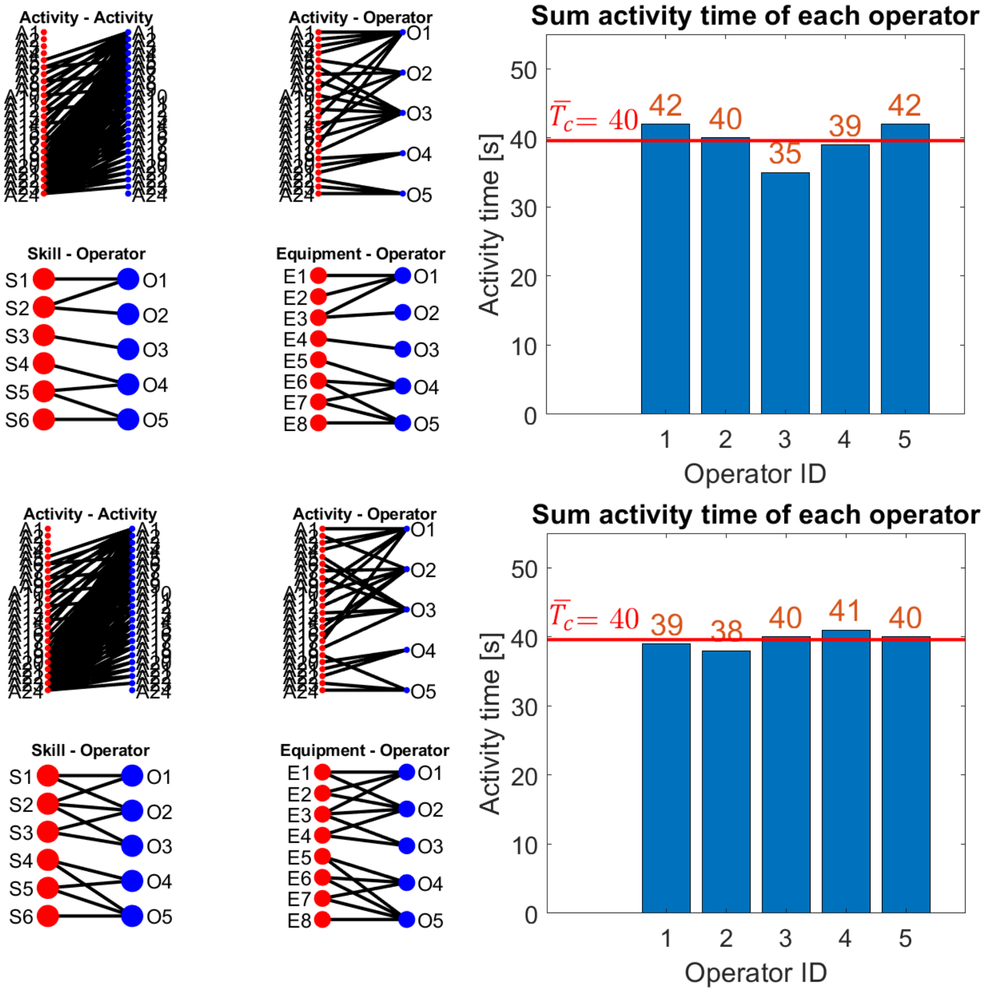


FIGURE II.2.9: Assembly line balancing result of the proposed method with time-skill-equipment cost included (on the top), and considering only time related cost (on the bottom side)

The second test is focusing on the sensitivity of the ALB solution to changes in activity times, and deviation if i.e. a specific operator performs a certain assembly activity with a delay. Figure II.2.10 shows the test of how an individual activity time deviation affects the cycle time. The figure represents three different assembly line balancing results with five operators, where in each case, one of the 24 activities has been performed for a longer than ideal time. In the first scenario activity 2 (connector handling) has been performed for 10 seconds, instead of the expected 3 seconds. This delay had a serious effect on the balancing as the time cost decreased to 77% and the mean activity time is 41 seconds. In the second scenario activity 16 (insertion 2^{nd} end) has been performed for 10 seconds, instead of the expected 5 seconds. This deviation caused an even better, 97% of time cost, while skill and

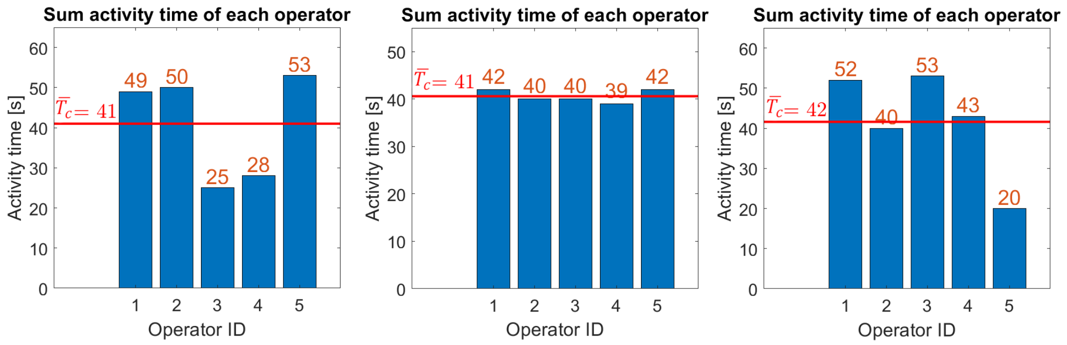


FIGURE II.2.10: Change of line balancing time results in the case of an activity has been performed for longer than ideal time, with a delay by the allocated operator - three different scenarios

equipment costs are still the same, 38% and 36%. Finally, in the third scenario activity 24 (QC final) has been performed for 20 seconds, instead of the expected 10 seconds. This delay had also a significant effect on the balancing as the time cost decreased to 78% and the mean activity time is 42 seconds.

In the third experiment, the assembly line balancing result has been tested with different number of operators (the original activity time list has been used, as listed in Table C.2 of Appendix C.1). Figure II.2.11 represents four scenarios of ALB result with three, four, five and six operators. By increasing the number of allocated operators, the mean cycle time is decreasing and the cost values are also changing. Table II.2.5 summarizes the cost results in further cases from two, to eight operators. Additionally, the mean cycle time, and the best cost of SA optimization algorithm (the weighted total cost of the multi-objective optimization function) are also listed in the table below.

TABLE II.2.5: Assembly line balancing result with different number of operators

Number of operators	Mean cycle time	Station-time-related objective	Skill-related (training) objective	Equipment-related objective	Best cost
2	99 sec	98%	43%	44%	38%
3	66 sec	90%	38%	40%	44%
4	50 sec	93%	38%	40%	50%
5	40 sec	94%	38%	36%	40%
6	33 sec	79%	38%	36%	49%
7	28 sec	66%	38%	40%	52%
8	25 sec	83%	30%	31%	52%

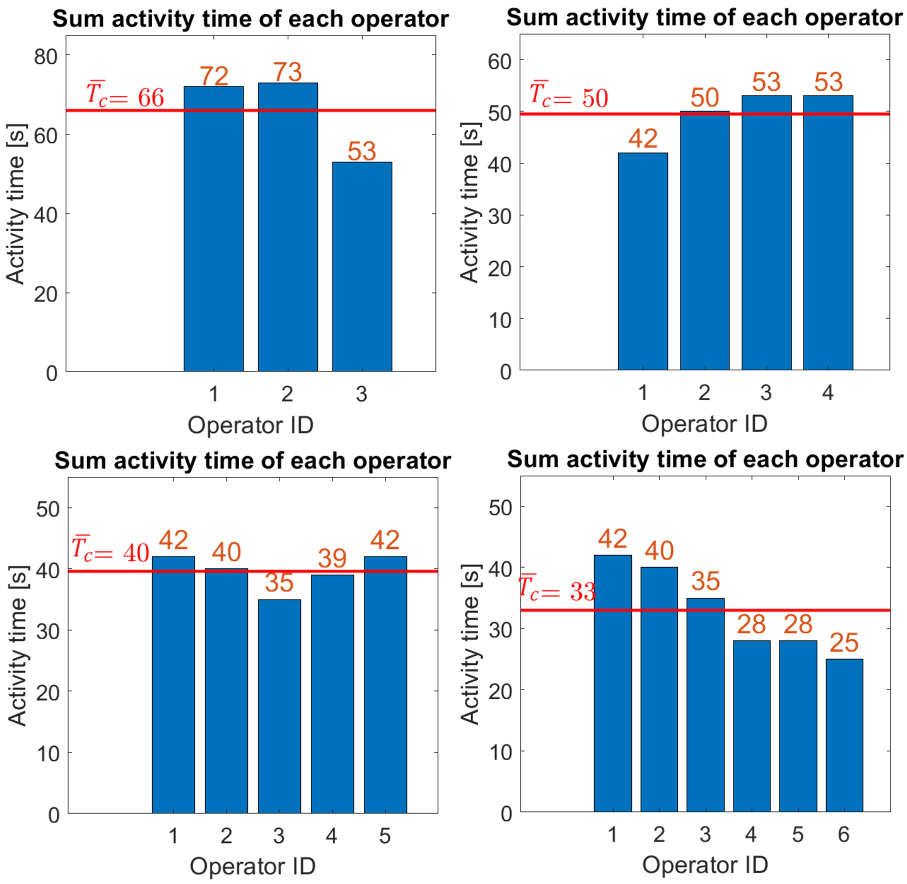


FIGURE II.2.11: The change of sum activity times and mean cycle time in the case of different number of allocated operators, three to six

Finally, Figure II.2.12 presents the different time, skill and equipment-related objectives in the case of different operators. As the results show, the increase in the number of operators decreases the efficiency of the utilization of the tools and skills (this trend is the main driving force for forming manufacturing cells). The process can be well balanced in the case of 3–5 operators; e.g., in the case of five operators, in one of the best solutions, the balancing objectives are a time cost of 94.3%, training cost of 75.0% and equipment cost of 72.8%.

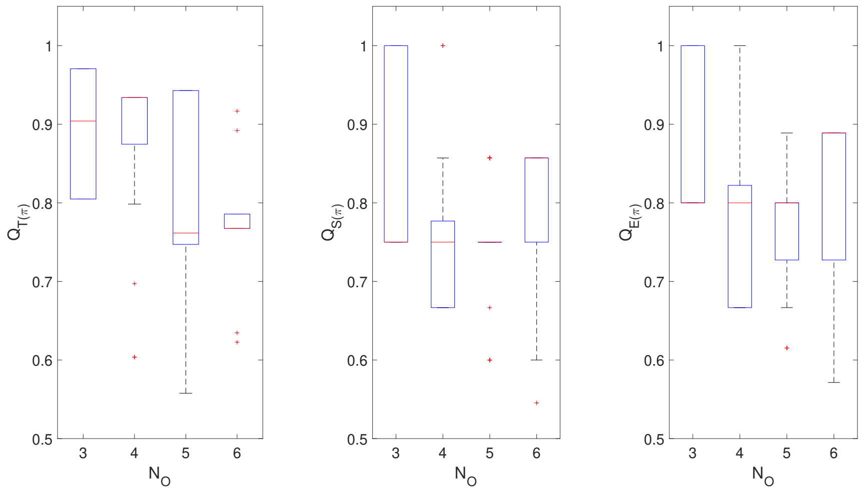


FIGURE II.2.12: Boxplot of time, skill and equipment-related objectives for different independent runs of the algorithm and with different numbers of operators

II.2.5 Complex, multilayer analysis of a wire-harness assembly graph network

This section describes the creation of RDF-based multilayer network to visualize and analyse the connectivity between the individuals of a production network. The collected manufacturing data is based on the same wire harness assembly case study presented in Appendix C.1 (Chapter C). After that the method is evaluated to discover the potential of community detection. The field of community detection in graph networks is presented in more detailed in the following Chapter II.3.

MuxViz [286] has been utilized to create multilayer graph representations and source other networks analyse. By graphically examining process networks, different inhomogeneities can be identified and analysed. Identifying the core nodes and investigating the critical edges or node-degree distribution within the network structure can provide internal information about the production process.

In this approach, the analysis scope is the skill, equipment, and workstation assignment in assembly activities. Figure II.2.13 shows the multilayer visualization, which contains three different layers, representing the connectivity of the 653 assembly activities to other classes of the ontology as Skills, pieces of Equipment and Workstations. Unique colours denote the core nodes of the network as five Skills (green), five pieces of Equipment (blue) and ten Workstations (red). Additional information is presented by internal edges within these three layers, connecting the core nodes and representing assignments as Workstation-Skill, Skill-Equipment and Workstation-Equipment.

The Activity-Workstation layer is investigated in more detail to uncover the possibility of merging activities within $w_3 - w_8$ stations. Figure II.2.14 presents the discovered communities based on multilayer connectivity, where workstations are classified into five communities ($\varphi_1 - \varphi_5$), which are listed in Table II.2.6. It can be concluded that the identified, highly related workstations are included in the same community based on several attributes. The multilayer analyses confirmed that merging these stations would be beneficial.

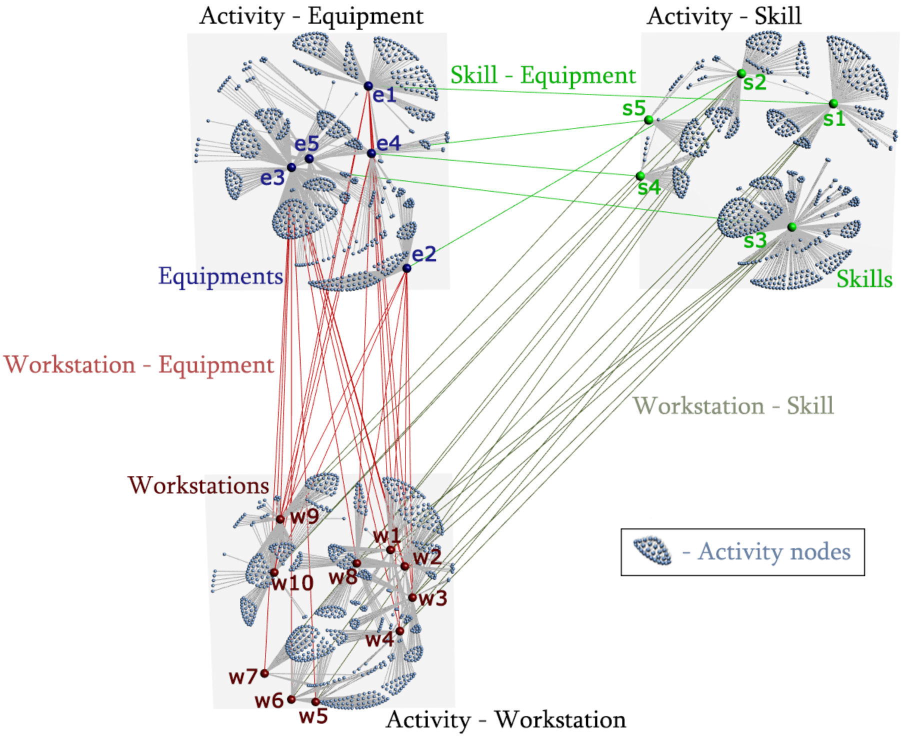


FIGURE II.2.13: Multilayer visualization of wire harness manufacturing data

TABLE II.2.6: The distribution of workstations into five different communities

Workstation	Community	Workstation	Community
w_1	φ_2	w_6	φ_4
w_2	φ_2	w_7	φ_4
w_3	φ_3	w_8	φ_5
w_4	φ_4	w_9	φ_1
w_5	φ_4	w_{10}	φ_1

In conclusion, the multilayer-based analyses can make the discovery of communities and critical elements of the production system more efficient, and it shows the

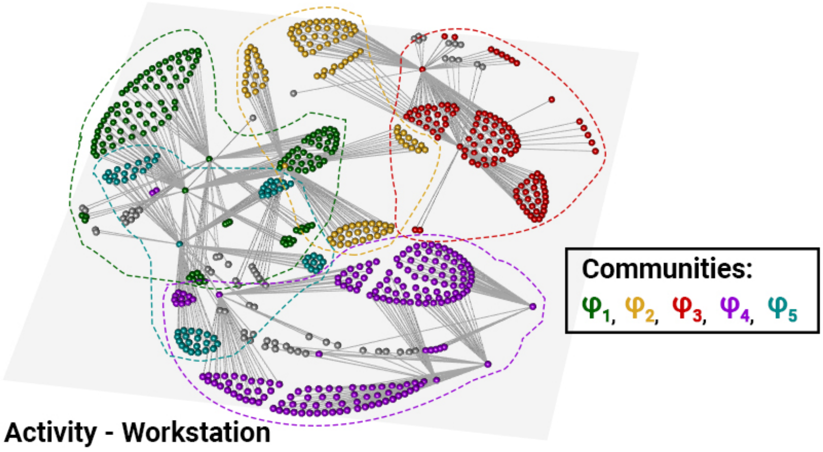


FIGURE II.2.14: Activity-Workstation layer and the identified communities

possibilities for process engineers to solve the line balancing problem considering all production parameters.

II.2.6 Summary of the proposed assembly line balancing method

An assembly line balancing algorithm has been proposed to improve the efficiency of production systems by the multi-objective assignment of tasks to operators. The optimization of this assignment is based on a multilayer network model that provides information about the activity times, constraints and benefits (objectives) of the assignments, where the layers of the network represent the skills of the operators, the tools required for their activities and the precedence constraints of their activities. The training and equipment costs as well as the precedence of the activities are also taken into account in the activity–operator layer of the network. As these costs and benefits are difficult to evaluate, the AHP technique is used to quantify the importance of the criteria. The optimization problem is solved by a multi-level SA algorithm, that efficiently handles the precedence constraints thanks to the proposed problem-specific representation.

The results show that the developed algorithm can be adapted to different scenarios, which has been represented by the comparative results, where the case study has been tested with a variety of different parameters. The scalability of the algorithm was not investigated, although it is already a well studied field in the case of simulated annealing algorithms [287, 288].

The AHP-based pairwise comparison of the importance of the nodes, edges and complex paths of this network can be used to evaluate the objectives of the optimization problems. The integration of the network-based knowledge representation and the AHP-based knowledge extraction makes the application of the proposed methodology attractive in complex optimization problems.

The developed algorithm was implemented in MATLAB and the applicability of the method demonstrated with an industrial case study of wire harness manufacturing. The results confirm that multilayer network-based representations of optimization problems in manufacturing seem to be potential promising solutions in the future. The main contribution of the work is that it presents tools that can be used for the efficient representation of expert knowledge that should be utilized in complex production management problems. The proposed multilayer network-based representation of the production line supports the incorporation of advanced (ontology-based) models of production systems and provides an interpretable and flexible representation of all the objectives of the line balancing problem.

Additionally it has been stated, that the multilayer graphs are suitable tools for analysing data stored in ontologies. Network and data science can support the analysis of complex systems represented by ontologies, and the multilayer-network-based analysis of ontologies supports production management.

Chapter II.3

Efficient network community detection algorithm based on crossing minimization and bottom-up segmentation

This chapter presents how communities in networks can be detected by integrating barycentric serialization-based co-clustering and bottom-up segmentation. Because the nodes are efficiently ordered according to their neighbors by barycentric serialization, the segmentation algorithm provides modules in a computationally more efficient manner than the most frequently used Louvain community detection algorithm. The efficiency of the method is compared with other community detection algorithms based on benchmark problems.

Considering the circumstances, communities in large networks are also aimed to find, and a community detection algorithm is presented that is based on the computationally efficient barycentric serialization of the nodes and bottom-up segmentation of the node orders. The key idea is that after the barycentric serialization of the adjacency matrix of the network, the detection of the communities can be considered as a segmentation where the segmentation cost is based on the modified cost function of the Louvain algorithm [289], which is the most frequently applied solution to handle the resolution limit problem [290]. This approach offers a robust solution that can provide results in a short time, even on large networks.

This chapter is structured as the following:

- Section II.3.1 presents the developed method in details as:
    - Modularity based analysis of communities in graph networks described in Section II.3.1.1.
    - The first module of the combined algorithm, the barycentric serialization-based crossing minimization, is discussed in Section II.3.1.2.
    - The second part of the algorithm, the community detection process, is presented in Section II.3.1.3, where the so-called bottom-up method is applied to perform modularity-based segmentation.
    - In Section II.3.1.4 the complexity analysis is discussed.
- Section II.3.2 evaluates the developed algorithm and presents several benchmark applications to prove the efficiency of the developed method.
    - First, subsection II.3.2.1 introduces the applied metrics for evaluation and the other algorithms for comparison.
    - In subsection II.3.2.2 the performance test results are presented.
    - Subsection II.3.2.3 discusses the merging process and the tuning operator test.

II.3.1 Crossing minimization and bottom-up segmentation based community detection method

This section describes the main steps and features of the combined algorithm. The base theory of modularity as a cost function is detailed in Section II.3.1.1. The serialization of the node adjacency matrix is discussed in Section II.3.1.2, that is based on barycentric coordinates and performed with crossing minimization. Therefore after that, the task is only to perform the segmentation of the already arranged nodes. In addition to that, the communities can be interpreted as segments. For the segmentation of the nodes, a bottom-up algorithm is applied (proposed in Section II.3.1.3), where the merging benefit of segmentation is equivalent to the value of modularity if the emerging modularity value of the new community is considered after the merging. The adjacent nodes (segments) are merged in the bottom-up algorithm to get the most favourable aggregation as the highest modularity value. The procedure merges the communities as long as the merged variant is favourable, which increases the modularity. The developed method provides a better performance, similar to other more complex algorithms, but at the same time, the algorithm has effective scalability, as presented in Section II.3.1.4.

II.3.1.1 Cost function - Modularity

This subsection discusses the theoretical background of network modularity Louvain algorithm.

Modularity is a measure of the structure of a graph or network, measuring the density of connections within a module or community. It was designed to measure the strength of the division of a network into modules (also called groups, clusters or communities). Networks with high modularity have dense connections between the nodes within modules but sparse connections between nodes in different modules. The modularity can be either positive or negative, with positive values indicating the possible presence of community structure. Modularity is commonly applied in optimization methods for detecting community structure in networks, where the optimization algorithm tries to detect communities in the graph or network based on their modularity [291].

Figure II.3.1 represents an illustrative example of four different scenarios of graph network (consist of 9 nodes and 13 edges) partitioning and also the related modularity values, where the color of the nodes represent the belonging to communities. Figure II.3.1-**a.** shows an optimal partition, where five nodes (green) form a commnunitiy in the network, and four others ones (purple) form another one. The modularity value of omptimally partioned network **a** is $0,41$. Figure II.3.1-**b.** shows and subomptimal network partition, where the nine nodes are grouped into communities in a different way, therefore the modularity value is only $0,22$. In the third scenario, in Figure II.3.1-**c.**, all nodes are part of the same single community, therefore the value of modularity is 0. Finally, Figure II.3.1-**d.** represents a scenario, where each nodes belong to different communities, which results a negative network modularity value $-0,12$.

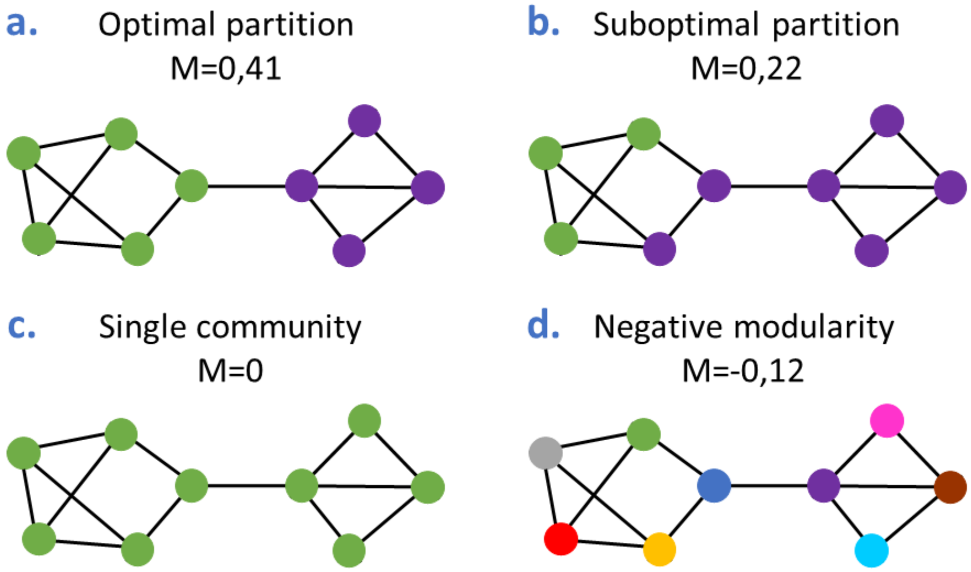


FIGURE II.3.1: Different type of graph network partitions and modularity values [291]

In summary, community detection algorithms aim to detect communities (partitions) in a network by maximising the modularity value of the network. Based on the literature review [248, 238], it can be stated that modularity is one of the most suitable optimization metrics in the community detection scenario.

The network is represented with a complete list of the links as $\mathbf{A}$ adjacency matrix, which is an undirected symmetric network, and the elements are $a_{i,j} = a_{j,i} = 1$ if the i and j nodes are linked and 0 if not. Consider a network with N nodes, L links, the size of $\mathbf{A}$ adjacency matrix is $[N \times N]$ and while partition them into C communities, in each C_c community $c = 1, \ldots, C$ having N_c nodes connected to each other by L_c links.

The connectedness quality of the communities is referred to as partitions and measured by the modularity of the partition. The iteration continues until no more change in the community network structure, while changes are observed and maximum modularity is reached. The Louvain algorithm does not require the number of communities as an input or their sizes before the execution. Louvain method consists of two parts, modularity optimization, and community aggregation, and both are performed until there is no change in the network and the maximum modularity is achieved. The high value of modularity means the network is partitioned well. Additionally, the modularity allows deciding if a particular community partition is better than other ones [292, 249].

Considering these, the formalised calculation of the modularity value is handled with equation (II.3.1), where M means the total modularity of a complex network, and M_c stands for the modularity of a given c-th community within the network. The total degree of the nodes in the C_c community is described with $k_c = \sum_{n=1}^{N_c} k_n$ [291], where k_n is the actual node degree. The variances are evaluate within the nodes of the network $(a_{i,j})$ and the expected number of links between i and j nodes if the network is randomly wired $(p_{i,j})$ as modularity (M_c). The probability of the randomly wired nodes with degrees k_i and k_j linking to each other is $p_{i,j} = \frac{k_i k_j}{2L}$ [291], and the modularity of the entire network is the sum of the modularity of communities:

$$M = \sum_{c=1}^{C} M_c = \frac{1}{2L} \sum_{(i,j)\in C_c} \left(a_{i,j} - \frac{k_i k_j}{2L} \right) = \sum_{c=1}^{C} \left[\frac{L_c}{L} - \left(\frac{k_c}{2L} \right)^2 \right] \tag{II.3.1}$$

When the L_c number of links in a node is larger than the expected number of links between the N_c nodes, then the nodes of the sub-graph C_c can be a part of a true community [293]. When M_c is positive, then the sub-graph C_c represents a potential community because it has more links than expected by chance. When M_c is zero, then the connectivity between the N_c nodes is random, fully explained by the degree distribution, while if M_c is negative, then the nodes of C_c do not form a community [291].

In the case of a directed network, the so-called Directed Louvain method [294] can be applied for modularity maximization. The behavior of the Louvain algorithm is the same in the directed case, the main difference is in the calculation of the gain of modularity obtained by adding vertex i to the community C [294]. In case of an weighted network the algorithm gets more complex with an additional input as a weight matrix describing the extents of interactions between the components of the network adjacency matrix [295]. The edge weights have to be also implemented into the modularity maximization algorithm as an additional cost of the function.

Important to mention and handle the so-called resolution limit of the modularity maximisation-based community detection methods. When the k_c total degree of the communities satisfies $k_c \leq \sqrt{2L_c}$, then modularity increases by merging two communities into a single community, even if these two communities are otherwise distinct [296, 291].

The novelty of the developed approach is that the adjacency matrix of the network (using crossing minimization) is serialized before the calculation of modularity, as it has to check only the neighbors. On the other hand, different methods need to analyse all edge pairs and node assignments. Thanks to this approach, the community detection procedure can be more efficient and faster than the basic Louvain algorithm.

II.3.1.2 Crossing minimization based serialization

This subsection introduces the serialization method, which aims to increase community detection efficiency, resulting in fewer iterations because, in a pre-serialized network, not all nodes have to be checked for modularity.

Graph network serialization has several forms and names in the literature, such as crossing minimization or bipartite clustering methods [297]. The crossing minimization algorithm has also been used for the so-called graph-drawing-based biclustering, which can successfully perform biclustering even with overlapping rows and columns in the presence of noise [298]. The crossing minimization reduces the number of edges between two sets of bipartite graphs by re-ordering the nodes, resulting in the 'similar' nodes being closer to each other. Moreover, the algorithm [217] provides an efficient co-clustering for bipartite graphs, where the barycentric ordering is applied during the crossing minimization. The algorithm arranges the rows and the columns of the adjacency matrix simultaneously. Because of the identical row orders, the barycenters are calculated as the weighted sum of the row barycenters of the individual matrices.

Figure II.3.2 represents a simple example of how the barycentric coordinates of bipartite graphs are formulated and how the node order can be serialized based on these values to reach fewer crossing edges in the graph.

The crossing minimization algorithm defines the ranked order of nodes as $\mathbf{x} = [x_1, \ldots, x_N]^T$ vector and calculates the $\mathbf{b} = [b_1, \ldots, b_N]^T$ barycentric coordinate vector as:

$$b_i = \frac{\omega_i \sum_{j=1}^{N} a_{i,j} x_j}{k_i} \qquad \forall\ i, \tag{II.3.2}$$

where ω_i is the weight of the i-th node.

In the method, random initialization is performed if there is no prior knowledge of the structure of the network, which may contain problem-relevant information. Also, each node of the network has a "order number". Furthermore, if there is no prior knowledge and the network is complex, random initialization with multiple runnings of the algorithm can help ensure that the method does not get stopped at a local optimum point. Multiple runnings are also required to handle this problem, as a random initialization itself may cause an uncertain result. In the case of a highly complex problem, random initialization of the nodes may be necessary for this purpose.

The HITS (Hyperlink-Induced Topic Search) centrality [299] has been applied to define the ω_i weight, which is a specific eigenvector technique for assessing the centrality of nodes in directed graphs [300]. The HITS algorithm, primarily used in search engines and web pages, distinguishes two types of web pages hubs and authorities. A node can be defined as an authority if many high-quality nodes link to it or a hub if it is linked to many high-quality nodes. Nodes (or pages in the case of web applications) pointing to a relevant node are likely to point as well to other relevant nodes, so to create a sort of bipartite structure where relevant nodes (authorities) are cited by particular nodes (hubs). Such bipartite structures allow the identification of the relevant pages for the user query [301, 302].

The algorithm (see in Algorithm II.3.1.2) is updating the value of $\mathbf{x}$, based on the barycentric coordinate values (Equation II.3.2) with the implementation of a rank correlation method [303]. The $\tilde{\mathbf{x}}$ vector is storing the $\mathbf{x}$ node order from the previous iteration. The utilization of these vectors is beneficial as the algorithm does not modify the original adjacency matrix ($\mathbf{A}$), ensuring fast and memory-effective implementation. Additionally, α is defined as a termination parameter of

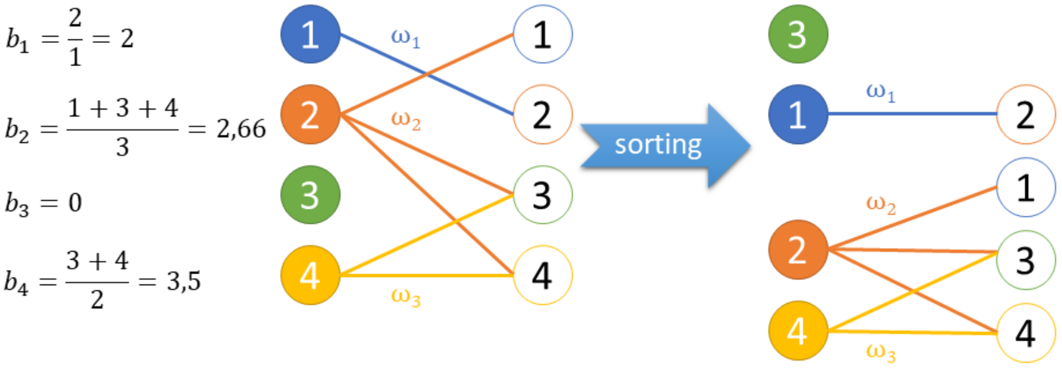


FIGURE II.3.2: Barycentric coordinates of bipartite graphs and node reordering with less edge crossing

Algorithm 2 Crossing minimization based serialization - Re-ordering with row-column changes based on barycentric coordinates

```
Initialisation: $\mathbf{A}, \mathbf{x}, \mathbf{b}, \alpha$
Initialisation for the loops: $iter, maxIter$
while ( $||\mathbf{x} - \tilde{\mathbf{x}}|| < \alpha$ )  OR  ($iter \leq maxIter$) do
    $\tilde{\mathbf{x}} := \mathbf{x}$ - store the order of nodes
    Calculate the $\mathbf{b}$ barycentric values based on the order of nodes ($\mathbf{x}$) - Equation II.3.2
    Calculate the rank order of the barycentric coordinates: $\mathbf{x}$
    $iter += 1$
Result: $\mathbf{x}$ as the ordered sequence of nodes
```

the algorithm, which aborts the iteration if the benefit decrease below this limit. The tuning of this stopping criterion ($\alpha > 0$) makes it possible to adjust the accuracy of the serialization, and the *maxIter* value is the maximal number of iterations.

II.3.1.3 Bottom-up segmentation based community detection

Thanks to the nodes are already serialized, the communities can be interpreted as segments of serialized time series, therefore the segmentation-based community detection (the second part of the developed method) is discussed in this subsection. The bottom-up algorithm is applied to merge the previously serialized nodes and segments while using modularity as a cost function. An advantage of this method is that the nodes are already serialized; therefore, the modularity test can be performed by evaluating which node is on the left and the right border of a specific community in the sequence. Thanks to this feature, the bottom-up algorithm can be applied very effectively.

The bottom-up algorithm can offer a solution for modularity optimization, as initially has been developed for time series segmentation in data mining and can support classification and clustering during the analysis of process data [304]. The principle of a bottom-up method is to start from the most satisfactory approximation, then continuously merge the segments (or communities) until one of the stopping criteria becomes true [305]. The bottom-up segmentation has a wide range of applications nowadays, as semantic segmentation based object detection [306], or optimizing a convolutional neural network with bottom-up clustering [307].

Segmentation algorithms simultaneously determine the parameters of the models used to approximate the behavior of the system in the segments and the borders of the segments by minimizing the sum of the costs of the individual segments [308], which is the $M = \sum_{c=1}^{C} M_c$ in this case.

The principle of the merging method is visualized in Figure II.3.3. On the vertical and horizontal axes, the serialized adjacency matrix is shown as the result of the crossing minimization, and also the potential communities are represented. During this merging procedure, data points are assigned to segments, therefore, the axes are duplicated as the *Communities* and the *Nodes*. An example for merge together two pre-arranged node communities is marked with $\Delta M_{c,c+1}$ (with light-blue color).

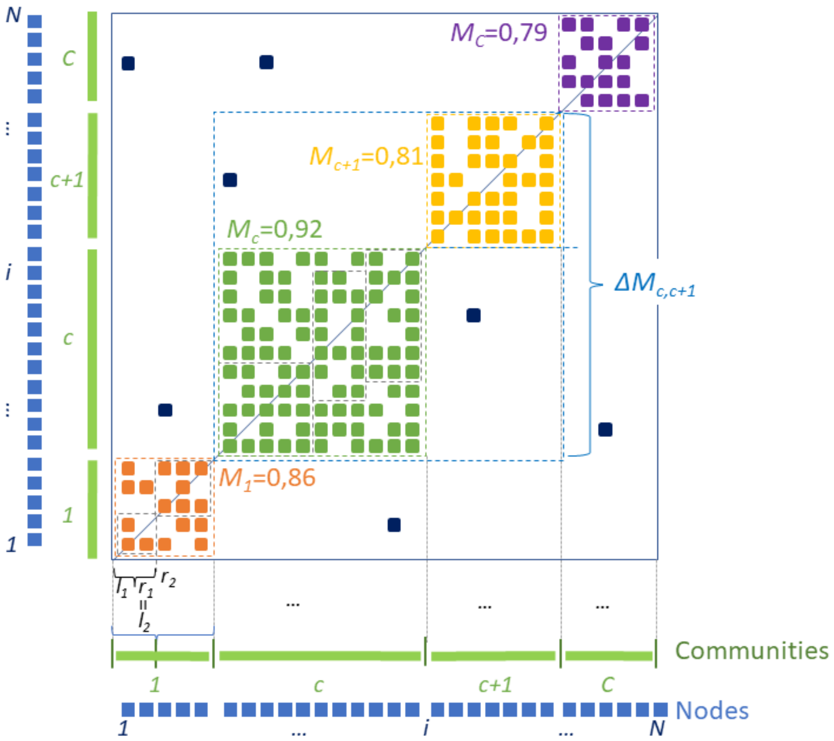


FIGURE II.3.3: Merging method of serialized node adjacency matrix based on modularity

The modularity of the merged communities (c-th and $c+1$-th) is described in Equation (II.3.3) based on the total modularity values (Equation (II.3.1)), where the total degrees of the merging communities are denoted by k_c and k_{c+1} [291, 309].

$$\Delta M_{c,c+1} = \left[\frac{L_{c,c+1}}{L} - \left(\frac{k_{c,c+1}}{2L}\right)^2\right] - \left[\frac{L_c}{L} - \left(\frac{k_c}{2L}\right)^2\right] - \left[\frac{L_{c+1}}{L} - \left(\frac{k_{c+1}}{2L}\right)^2\right] \tag{II.3.3}$$

Furthermore, $L_{c+1} = L_c + L_{c+1} + l_{c,c+1}$ and $l_{c,c+1}$ is the number of direct links (edges) between the nodes of communities c and $c+1$ (L_c and L_{c+1} are the total number of links within the communities), and $k_{c,c+1} = k_c + k_{c+1}$ is the final degree of the merged communities.

The so-called Potts and the RB (Reichardt and Bornholdt) method has been applied to handle the resolution limitation [310, 311]. The $\gamma > 0$ value is implemented as a tuning operator to avoid the resolution limit problem and works as a threshold to filter out the weak connections between the segments, that aimed to merge.

The simplified formula of the modularity change after merging with the γ tuning operator is the following:

$$\Delta M_{c,c+1} = \frac{l_{c,c+1}}{L} - \gamma\frac{k_c k_{c+1}}{2L^2} \tag{II.3.4}$$

The crucial step of the segmentation is to find the border elements of individual communities in the barycentric coordinate-based serialized adjacency matrix. Boundaries of the segments (communities) are defined as l and r, herewith l_c is the left and r_c is the right segment boundary of the c-th community and the following community (in the sequence) will be bounded by $l_{c+1} = r_c$ (or $l_c = r_{c-1}$).

The cost function gives back a change of the modularity value after merging two segments of nodes, which have been pre-serialized, so the modularity value of the nodes within the start and the end node of the specific cluster l_c and r_c. If the algorithm merges the c-th segment with the next, $c+1$-th, than the modularity change will be $cost(l_c, r_{c+1})$. The *mergecost* function defines the difference between the merged modularity and the two individual modularity, therefore $mergecost_c = cost(l_c, r_{c+1})$. The goal is to monitor the rate of changes in modularity and detect if there is a significant improvement in the case of merging segments. Considering these, the *mergecost* is the change of modularity after merging segments will be:

$$mergecost_c = cost(l_c, r_{c+1}) = \Delta M_{c,c+1} - (M_c + M_{c+1}) \tag{II.3.5}$$

In general, there is no guarantee that serialization will always provide a better modularity basis for graph segmentation-based community detection algorithms. The effectiveness of serialization can depend on various factors such as the specific algorithm, the network structure and the used parameter settings [247]. Studies of various community detection algorithms showed that serialization can improve the modularity of the resulting partition, but note that the effectiveness can depend on the specific algorithm and network structure [246, 312]

Important to mention the philosophy of setting the size of the initial segments. The pre-serialized adjacency matrix is split into equal parts in the first step. If each node is an initial starting community, the merging cost of their vicinity is not necessarily positive. Therefore, it is advisable to set the size of starting blocks equivalent to the value of l_c as the 'lower limit'. The larger the division of community size, the faster the algorithm, but the less efficient it can find the highly connected nodes as segments. Furthermore, the initial community size also determines the *resolution*, in case of a too large value, it can be assumed that the actual community border will be inside the starting block. If there is no information about the tested network, equally distributed initial module sizes are used and $resolution = N/1$, which provides the most accurate result, but with a high number of iterations. In other cases with large networks, the *resolution* ratio is increased, and the runtime becomes shorter.

Algorithm II.3.1.3 describes the second part of the developed method.

Algorithm 3 Bottom-up segmentation based on modularity

```
Input: The boundaries ($l_c$ and $r_c$) based on the re-ordered adjacency matrix with crossing minimization (based on the $\mathbf{x}$ from Algorithm II.3.1.2)
Calculate the initial modularity ($M$) of the segmentation costs, based on Equation II.3.1
Calculate the merging costs of each community ($mergecost_m$) based on Equation II.3.5
while $max(mergecost) > 0$ do
    Find the best pair to merge: $p = argmax_m(mergecost_m)$
    Merge the two communities and recalculate the merging costs
    $mergecost_p = cost(l_p, r_{p+1})$
    Update $M, C, l_p$ and $r_p$
```

II.3.1.4 Complexity analysis of the algorithm

Due to the fact that an NP-complete problem [313] is investigated, the analysis of the computational complexity [314] has high importance in the case of community detection algorithms. This subsection summarizes the main characteristic and efficiency of the most relevant community detection algorithms. Furthermore, the computational performance of the developed method is also discussed.

Several approaches have been compared for community structure identification, where the correlation between the accretion of the models and the cost of the computation [315]. The complexity of the classical barycenter technique is $\mathcal{O}(|N| + |L|logL)$ [297]. The algorithm solves the crossing minimization rapidly, and similarly to, the multilayered application of crossing minimization usually stops after $5 - 10$ iterations [297]. The procedure calculates the node ranks in each set of nodes linear algebraic way with matrix and vector multiplication. It should be noted that matrices describing the cell formation problem are sparse so that they can be stored and handled efficiently [297].

In case of Ravasz [316], Girvan-Newman [317], and Greedy Modularity [242] the computational complexity of the algorithm is $\mathcal{O}(N^2)$, with Louvain [244] the complexity is $\mathcal{O}(L)$. The Louvain algorithm is more limited in case of storage demand than by computational time [291, 315].

The benefit of the developed algorithm depends only on the nodes, so it is not needed to get through on every edge (L), therefore instead of the $\mathcal{O}(L)$ the complexity is $\mathcal{O}(N)$.

II.3.2 Results and discussion of the developed combined algorithm

This section discusses the test results of the developed, combined algorithm. First, subsection II.3.2.1 introduces the applied metrics for evaluation of community detection performance and the other algorithms, which are used for comparison. In subsection II.3.2.2 the performance test results are discussed, where the developed method is compared with other algorithms on several benchmark networks. Finally, subsection II.3.2.3 presents the merging process and the gamma value testing of the developed algorithm.

II.3.2.1 Details of the applied metrics and other algorithms to compare

The developed algorithm has been tested on several benchmark networks, which are summarized in Appendix C.5 in Table C.12 (Chapter C). The test networks are coming from different studies, related to network science, community detection methods [318, 319], and from the *networkrepository.com* webpage [320].

In order to test various community detection algorithms, Girvan and Newman firstly gave an artificial network [317], called GN benchmark. Due to its simple structure, most community detection algorithms perform very well on the GN benchmark. Although, standard benchmarks, like that by Girvan and Newman, do not account for important features of real networks, as the fat-tailed distributions of node degree and community size. Therefore, LFR (Lancichinetti-Fortunato-Radicchi) benchmark graphs offer a solution, in which the distributions of node degree and community size are both power laws, with tunable exponents [234]. In a LFR benchmark graph, the degree of nodes and the size of communities obey the power law distributions. Additionally, there are several parameters involved in LFR benchmark, among them, N is the total number of nodes, $< k >$ and k_{max} are the average degree and maximum degree, respectively. $m(min)$ and $m(max)$ denote the minimum and maximum community size. The parameter μ represents the ratio of the external degree of each node. Obviously, with the increase of the mixing parameter μ, the community structure of LFR network is more indistinct.

In the experiments, the community membership of the nodes is compared with a so-called reference community list, that contains the memberships of each node as integer label numbers starting with 1. A big advantage of LFR benchmark networks is that the graph-generating algorithm provides not only the edge list of the network but this reference community list as well.

The calculation of the modularity value during the benchmark is performed with the classical Newman-Girvan modularity evaluation [249], based on the detected final communities and how the serialized nodes are segmented into modules. Additionally, evaluation metrics for testing besides modularity are the followings:

- **NMI** (Normalized Mutual Information) [321]:
  This criterion is calculated for measuring the similarity between two clusters. For a given network with N nodes, the NMI value between two divisions $X = \{X_1, X_2, ..., X_{m(X)}\}$ and $Y = \{Y_1, Y_2, ..., Y_{m(Y)}\}$ can be defined as:

  $$NMI = \frac{-2\sum_{i=1}^{m(X)}\sum_{j=1}^{m(Y)} n_{ij} log(\frac{n_{ij}\cdot N}{n_{Xi}\cdot n_{Xj}})}{\sum_{i=1}^{m(X)} n_{Xi}\cdot log(\frac{n_{Xi}}{N}) + \sum_{j=1}^{m(Y)} n_{Xj}\cdot log(\frac{n_{Xj}}{N})}. \tag{II.3.6}$$

  In the above equation, $m(X)$ and $m(Y)$ denote the community numbers of partitions X and Y, respectively, n_{ij} is the number of common nodes in communities X_i and Y_j. For the variables $W = \{n_{X1}, X_{X2}, ..., n_{Xm(X)}\}$ and $Z = \{n_{Y1}, n_{Y2}, ..., n_{Ym(Y)}\}$, n_{Xi} and n_{Yj} represent the numbers of nodes in X_i and Y_j. The denominator of NMI is just the sum of the entropies of W and Z. Note that the value of NMI is in the range $[0, 1]$ and equals 1 only when two community divisions are exactly consistent.

- **ARI** (Adjusted Rand Index) [322]:
  Adjusted rand index is based on pair counting and computed as follows:

  $$ARI = \frac{\sum_{i=1}^{m(X)}\sum_{j=1}^{m(Y)}\binom{n_{ij}}{2} - \Omega}{\left[\sum_{i=1}^{m(X)}\binom{n_{Xi}}{2} + \sum_{j=1}^{m(Y)}\binom{n_{Yj}}{2}\right]/2 - \Omega}, \tag{II.3.7}$$

  where Ω is given by $\Omega = \sum_{i=1}^{m(X)}\binom{n_{Xi}}{2}\cdot\sum_{j=1}^{m(Y)}\binom{n_{Yj}}{2}/\binom{N}{2}$.

Modularity reflects the closeness of the internal connection of the community through the difference between the strength of the connected edges in the actual community and the strength of the connected edges in the network under random division. NMI and ARI indicate the accuracy of community detection mainly by comparing the consistency between the results of community detection and the "true" community division (reference community list). The larger the NMI and ARI values, the better the effect of community detection is. The values of NMI and ARI are in the range $[0, 1]$ and equals 1 only when two community divisions are exactly consistent.

The efficiency of the presented method was compared with other modularity maximization procedures, listed in Table II.3.1. In each case, the input of the algorithms is the adjacency matrix of the graph.

Table II.3.1: List of the benchmark algorithms, using modularity maximization

	Short method description	Ref.
GCDanon	Danon's greedy community detection agglomerative method.	[323]
GCModulMax3	Newman's greedy agglomaerative modularity maximization method.	[324]
GCModulMax2	Fast greedy modularity maximization method.	[242]
GCModulMax1	Method of Blondel, Guillaume, Lambiotte and Lefebvre.	[244]
GCReichardt	Gamma-modularity maximization.	[236]

II.3.2.2 Comparing the performance of the algorithm with other methods

This subsection discusses the result of the algorithm test, using several performance metric and several type of networks.

In the first case such networks have been tested, which have available reference community lists as well, and thanks to that NMI and ARI values are computable. Table II.3.2 contains the parameters of three LFR networks, with different mixing parameter (μ) values. In Table II.3.3 details of real-life networks can be found, which examples have also a reference community list, in the case of the Lesmis network it is based on [325], and in the case of the Karate network it is based on [326].

Table II.3.2: The parameters of the LFR networks: node number (N), edge number (L), average degree ($< k >$), maximum degree ($< k_{max} >$), minimum community size as number of nodes ($m(min)$), maximum community size as number of nodes ($m(max)$), the ratio of external degree of all nodes (μ), and community number (m), based on the reference community list

	N	L	$< k >$	k_{max}	$m(min)$	$m(max)$	μ	C
LFR-1	1.000	19.922	20	50	20	100	0,1	24
LFR-2	1.000	19.420	20	50	20	100	0,4	19
LFR-3	1.000	19.898	20	50	20	100	0,75	24

Table II.3.3: The statistical properties of the real-life networks, which include: node number (N), edge number (L), average degree ($< k >$), clustering coefficient ($< c >$), average path length ($< d >$) and community number (m), based on the reference community list

	N	L	$< k >$	$< c >$	$< d >$	C
Karate	34	78	4.558	0,588	2.408	2
Lesmis	77	254	6.597	0,736	2.641	8

Figure II.3.4 represents a benchmark test of NMI, ARI and modularity applied on the LFR networks of Table II.3.2 and real-life networks of Table II.3.3. Figure II.3.5 represents the efficiency of detecting the right number of communities, applied on the LFR networks of Table II.3.2 and real-life networks of Table II.3.3.

Based on the result of Figures II.3.4 and II.3.5 the following conclusions can be made:

- In the case of LFR networks the developed algorithm performs lower modularity value, but can detect the number of communities accurately (as shown in Figure II.3.5), and also get better NMI and ARI.
- If the μ mixing parameter is large $(0, 75)$ all six algorithms have a low performance in NMI and ARI metrics, which is resulted by the very noisy networks structure.
- As the value of μ is increasing in LFR networks, as worse is the accuracy of the other five methods in finding the right number of communities. In case $\mu = 0, 75$ only the developed algorithm is able to find the original 24 communities, while the other methods find only 6-11 communities.
- In the case of the Karate and the Lesmis, real-life networks the NMI and ARI values are significantly better than others, while the value of the modularity is in the same range. Furthermore, only the developed method can find the exact number of original communities, 2 and 8.

Figure II.3.6 shows one benchmark result from the test, where the sub-plots represent the original adjacency matrix of the network, the serialized node structure, furthermore the *Modularity* and *Runtime* results. The detected communities are also highlighted on the serialized network. In this example, the network contains 4941 nodes, and after the crossing minimization based serialization, the bottom-up algorithm performs 451 merges to get 43 final communities. The parameters were

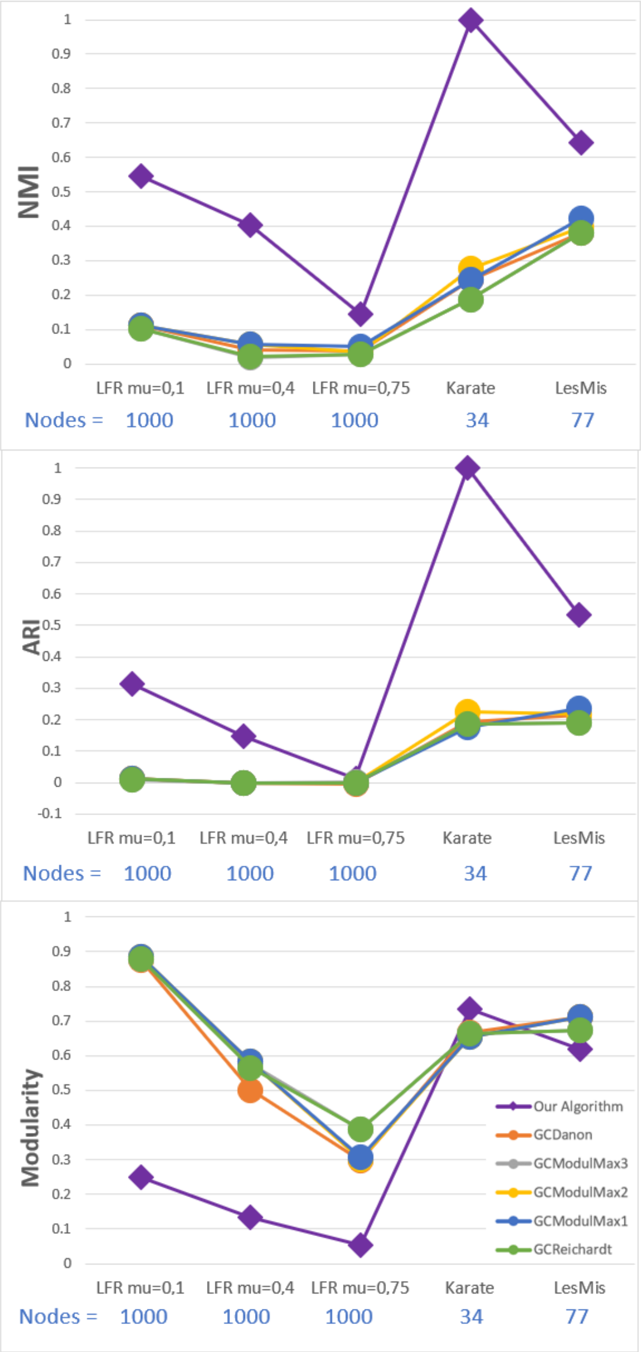


FIGURE II.3.4: Comparison of NMI, ARI and Modularity results of the algorithms, using two real-life networks and three different LFR networks

$\gamma = 6$ and $resolution = N/10$. It can be noted that the developed algorithm has a significantly faster *Runtime*, while the *Modularity* value is still in the same range as other methods. The runtime of the developed algorithm is $0,21$ second, the second fast method, the *GCModulMax2* did run for $4,82$ seconds, the *GCModulMax1* finished in 119 seconds, while each other methods have a runtime above 500 seconds.

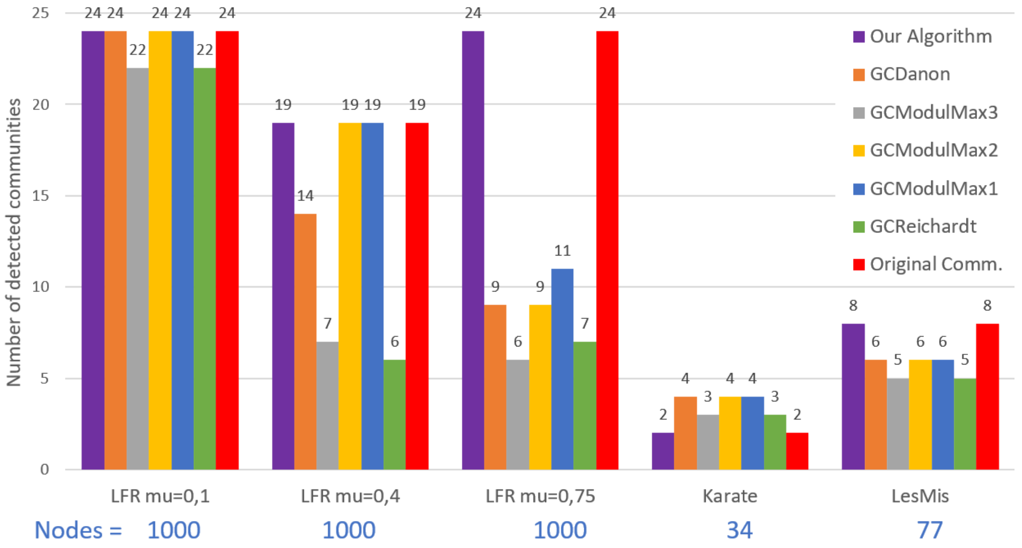


FIGURE II.3.5: Comparison of the number of found communities, compared with the original (red) reference, using two real-life networks and three different LFR networks

In the followings the test focus on variety type of real-life and generated networks with different node numbers. The value of modularity, the number of detected communities, and the runtime results have been analysed on different benchmark networks. Table C.12 (in Appendix) summarizes the test results on real-life and artificial networks. Figure II.3.7 visualizes the result of real life networks, below 500 nodes and Figure II.3.8 shows the result of benchmark networks, above 1.000 nodes. Most methods take significantly more runtime to finalise the community detection procedure if the network contains more than $\sim$ 500 nodes, as it is visualized on the bottom side of the figure, where the time axes are on a logarithmic scale.

Figure II.3.9 visualizes a test on generated network [327], where the size of the network has been increased from 500 nodes with 400 steps, till 2.900 nodes. Other parameters of the generated networks are 6 modules, $0,5$ total link density, and 20 ratio of the nodal degree to nodes within the same module. Each procedure detected 6 communities with each network size therefore, only the modularity values and the runtimes are visualized. The developed method was applied with $N/3$ resolution and $\gamma = 10$ in all scenarios.

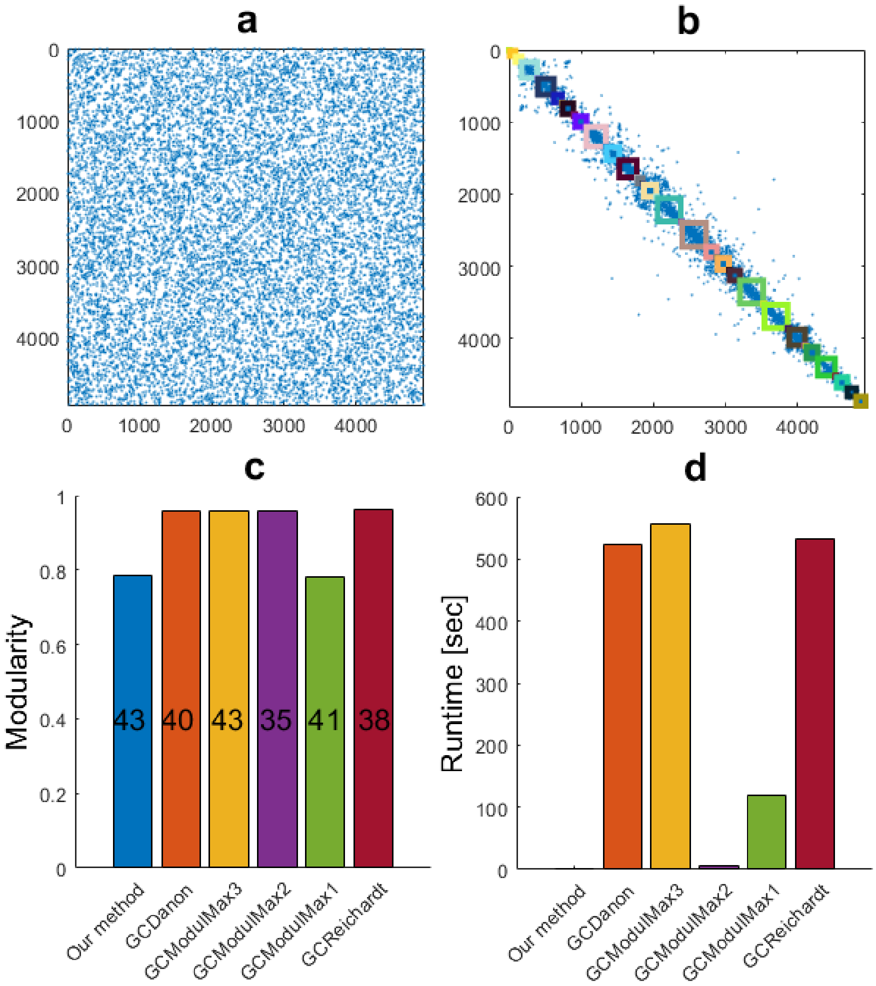


FIGURE II.3.6: Benchmark on *infpower* network, with 4941 nodes, visualizing the Original adjacency matrix (a), the serialized adjacency matrix with the detected communities (b), the number of detected communities, and their modularity values in the case of the six different algorithms (c), and the runtime of the six different community detection procedures (d)

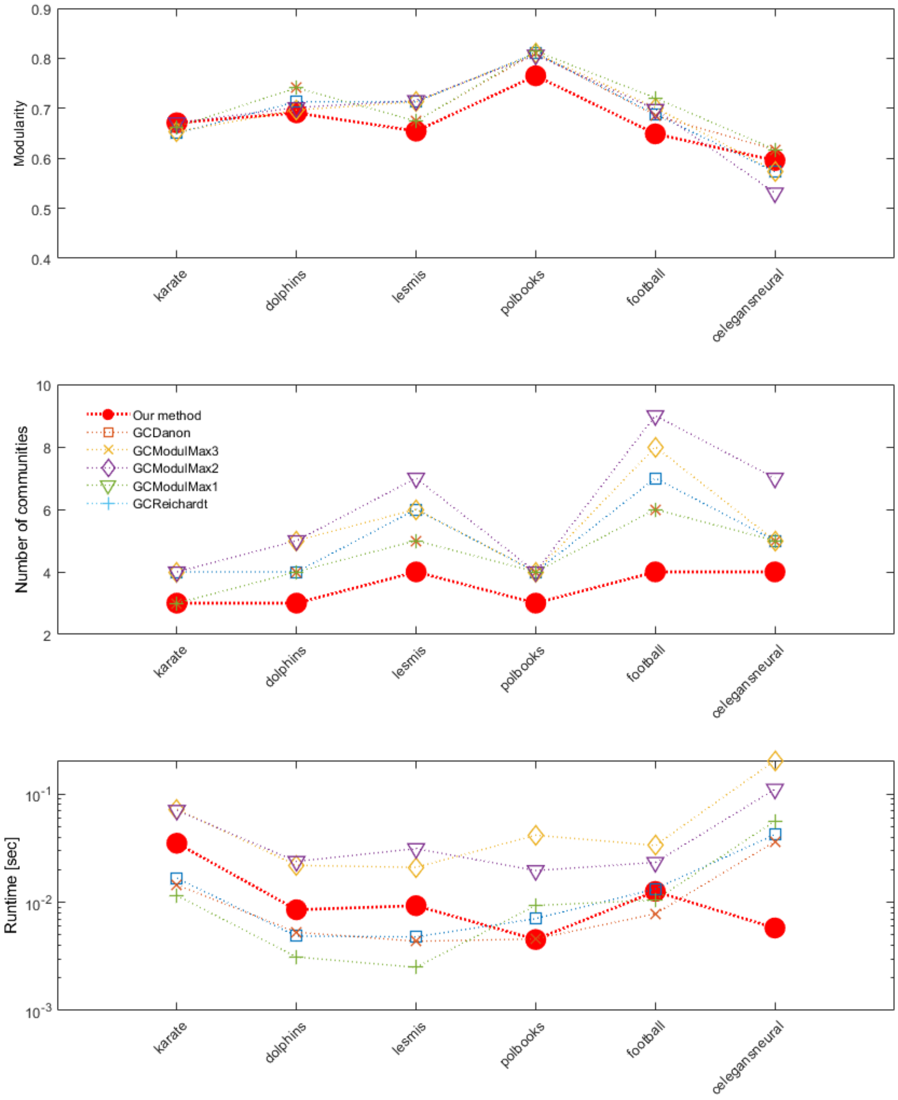


FIGURE II.3.7: Modularity (top), number of detected communities (middle) and the runtime on logarithmic scale (bottom) results of 6 different real life networks: **karate** - Zachary's karate club [328] (34 nodes and 2 official communities), **dolphins** - Dolphin social network [329] (62 nodes and 2 official communities), **lesmis** - Les Miserables [330] (77 nodes), **polbooks** - Books about US politics [331] (105 nodes and 3 official communities), **football** - American college football [317] (115 nodes and 12 official communities), **celegansneural** - Neural network [332, 333] (297 nodes)

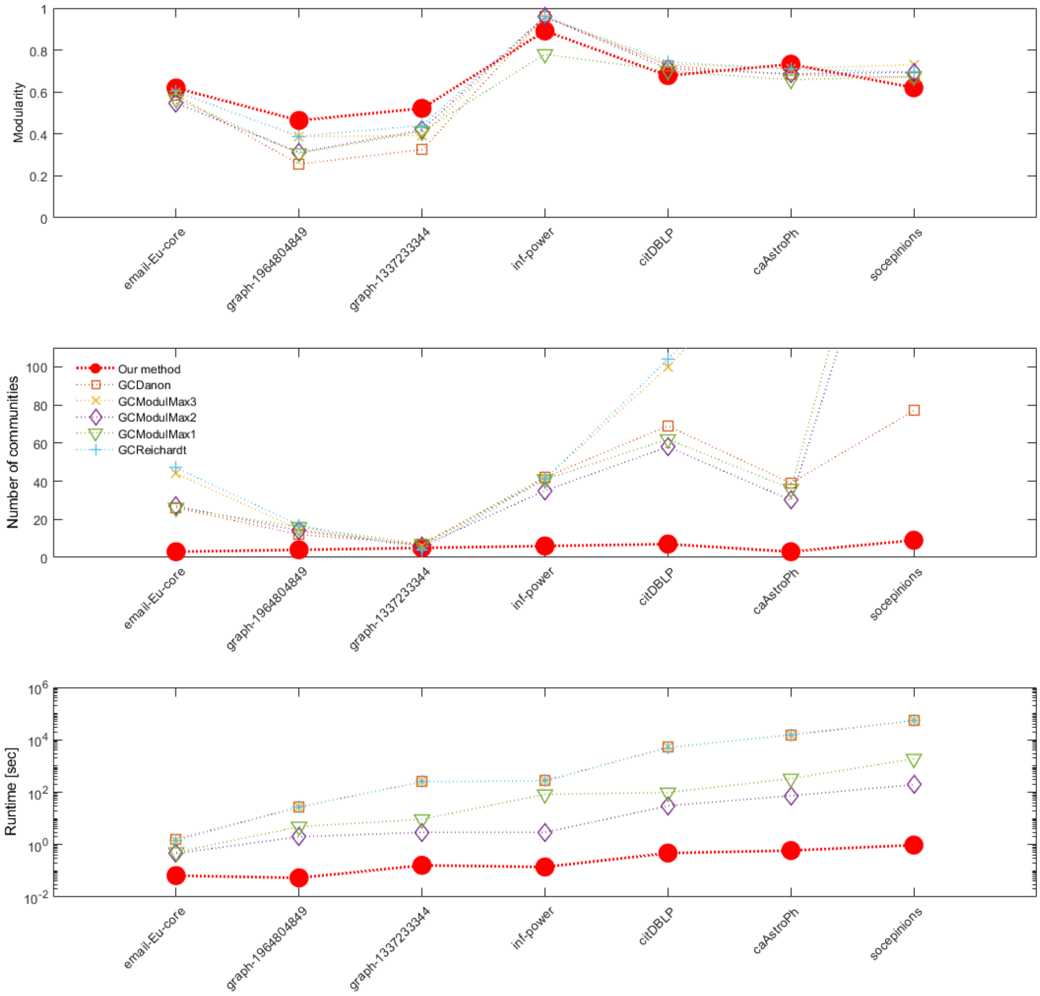


FIGURE II.3.8: Modularity (top), number of detected communities (middle) and the runtime on logarithmic scale (bottom) results of 7 different networks: **email-Eu-core** - Email flow between research institution members [334, 335] (1.005 nodes), **graph-1964804849** [318] (2.400 nodes), **graph-1337233344** [318] (4.800 nodes), **inf-power** - Western States Power Grid of the United States [333, 320] (4.941 nodes), **citDBLP** - Citation Network [320] (12.591 nodes), **caAstroPh** - Astro Physics collaboration network [334, 320] (17.903 nodes), **socepinions** - Epinions social network [336, 320] (26.588 nodes)

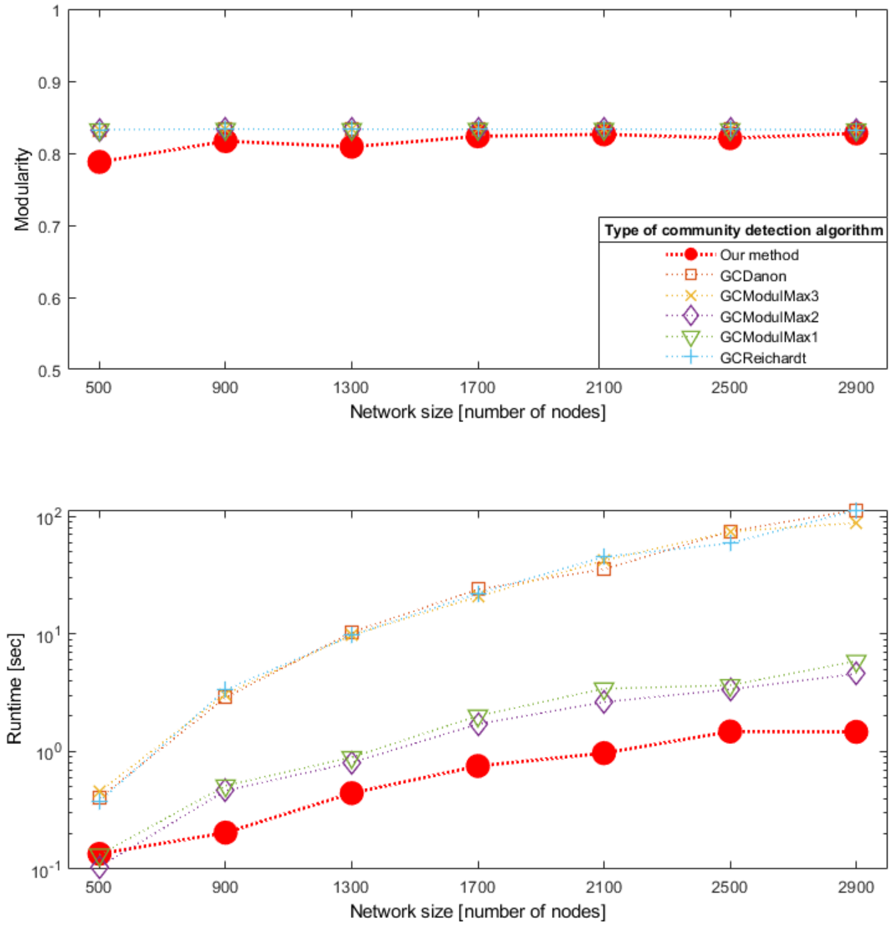


Figure II.3.9: Modularity (top) and the runtime (bottom) test results of the generated network, with seven different network sizes and six communities

II.3.2.3 Merging process and gamma value testing of the developed algorithm

This subsection presents the merging process of the developed algorithm, and also shows a sensitivity test for γ value, which is the tuning operator to handle the resolution limit problem.

Figure II.3.10 represents the hierarchical grouping process in the bottom-up segmentation, where it is stored, which segments are merged and in which iteration. Demonstrating the philosophy, a different benchmark network has been applied in this case, with 105 nodes and 20 initial segments. The procedure performs 16 iterations to get the 4 final communities, and the final network modularity is 0.71. The serialized adjacency matrix is visualized on the left, with the four detected community blocks. On the right of the figure, the merging process is visualized with a dendrogram, where the horizontal dimension is proportional to the increment steps of the modularity value after merging communities together.

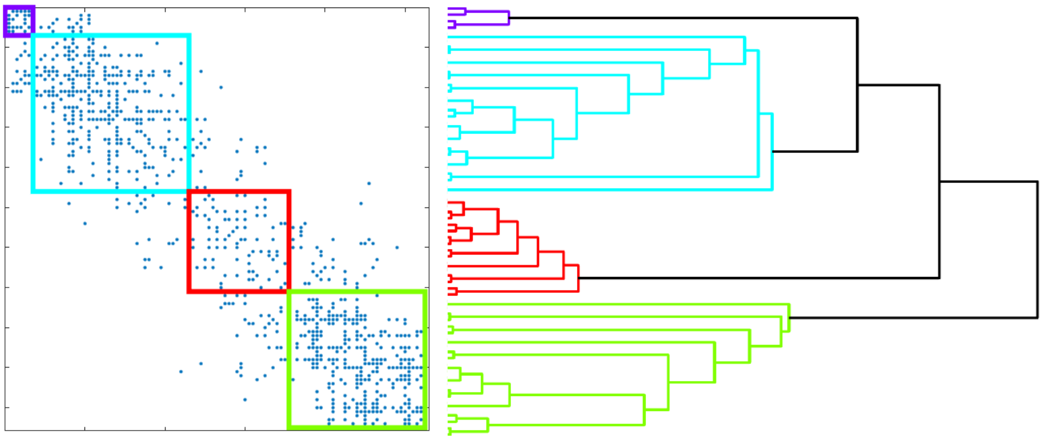


Figure II.3.10: Detected communities on the serialized adjacency matrix (left) and a dendrogram of the iteration steps while merging communities - Books about US politics network (105 nodes)

On Figure II.3.11 a test example is visualized about how many final communities could be find with different γ values. This test has been performed on a generated network [327], where the parameters are the followings: 500 nodes, 6 modules, $0,5$ total link density, and 20 ratio of the nodal degree to nodes within the same module. The result of the γ test highlights that after a particular value of γ, the algorithm rapidly resulting significantly more detected communities and then sets this amount as a plateau maximum.

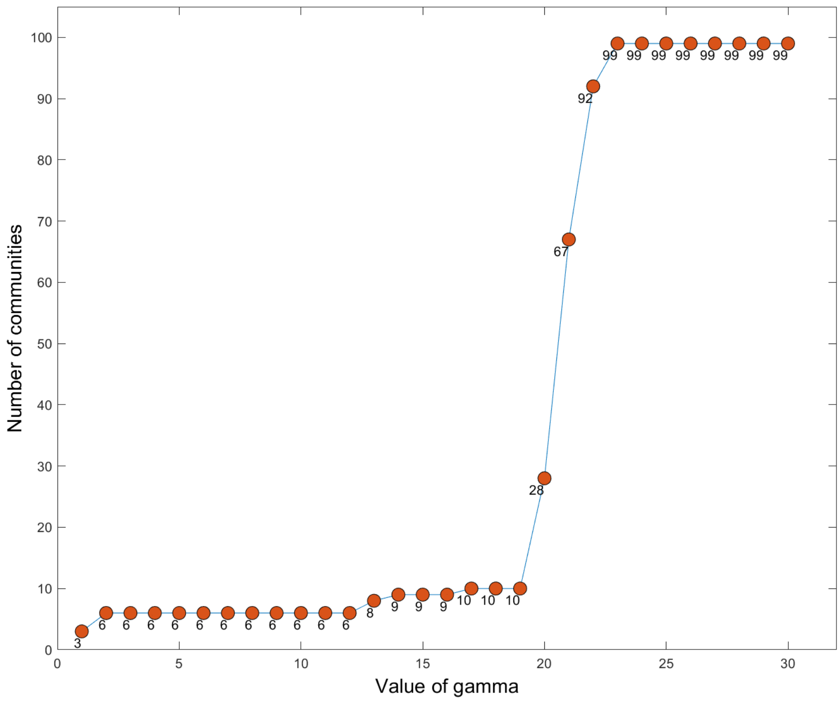


FIGURE II.3.11: Number of detected communities with different gamma values, where the curve shows the lower- , the upper plateau, and also the ramp-up section - Generated network with 6 modules (500 nodes)

II.3.3 Summary of the developed network community detection method

In this chapter, a modularity-based community detection method has been developed, using the combination of barycentric serialization-based crossing minimization and bottom-up segmentation. The developed method can perform scalable and time-efficient community detection compared with other Louvain algorithm-based procedures. Thanks to the pre-serialization of the adjacency matrix, the method has reduced iteration cost, as it does not need to test the entire data set.

This work highlights that the integration of barycentric serialization with modularity maximisation based bottom-up segmentation offers an efficient solution, especially on large data sets (above $\sim$ 1.000 nodes), which has been proved on benchmark problems using real-life, and generated networks. Additionally, the developed method is able to detect the right number of communities in the network, compared with a reference list. The study contains a test about the γ tuning operator that aims to investigate the resolution limit of the algorithm, and the number of the detected communities, with different setpoints. Furthermore, clustering and dendrogram visualization about the detected communities have been created.

Chapter II.4

Hypergraph-based analysis of collaborative manufacturing

This chapter aims to present a hypergraph-based analysis method. The design of these Operator 4.0 solutions requires a problem-specific description of manufacturing systems, the skills, and states of the operators, as well as of the sensors placed in the intelligent space for the simultaneous monitoring of the collaborative work. The design of a collaborative manufacturing requires the systematic analysis of the critical sets of interacting elements. The proposal is that hypergraphs can efficiently represent these sets, moreover, studying the centrality and modularity of the resultant hypergraphs can support the formation of collaboration and interaction schemes and the formation of manufacturing cells.

The main finding of this chapter is that the development of these solutions can be applied in collaborative manufacturing. Collaborative manufacturing aims to achieve real-time monitoring-based control for semi-automated production systems, thereby creating more precise collaboration between human workers and machines. The key idea is that hypergraphs can efficiently represent these sets, moreover, studying the centrality as well as modularity of the resultant hypergraphs can support the formation of collaboration and interaction schemes in addition to the creation of manufacturing cells.

This work aims to analyse how hypergraphs can be used to design collaborative manufacturing. According to this aim, the main contributions and structure of this chapter are as follows:

- The problem formulation and the background of collaborative manufacturing is discussed in Section II.4.1.
- The methodology of higher-order network representation to facilitate collaboration is described in Section II.4.2.
  - Firstly, in Section II.4.2.1 hypergraph-based modeling of complex manufacturing systems is presented.
  - In Section II.4.2.2, the main principles of hypergraphs are discussed.
  - A simple example of hypergraph-based modeling and theoretical visualization of a production process is given in Section II.4.2.3.
  - Finally, in Section II.4.2.4, the hypergraph centrality measures are proposed to identify the key elements and relationships according to collaborative manufacturing.
- A hypergraph-based case study of collaborative manufacturing and the discussion of the results are presented in Section II.4.3.
  - In Section II.4.3.1, several representation methods of the hypergraph-based wire harness manufacturing model are shown.
  - Section II.4.3.2 describes the identification methods of the critical elements and collaboration scenarios.
  - In Section II.4.3.3, the segmentation analysis of the production process is discussed.
  - Section II.4.3.4 summarizes the benefits of hypergraph-based analysis and discusses some further application possibilities.
- Finally, in Section II.4.4 the types of information that can be extracted from the network analysis and how those can be utilized for the redesign of manufacturing systems are concluded.

II.4.1 Collaborative manufacturing

Today, as manufacturers struggle with shortages of highly skilled personnel, the value of effective collaboration is more significant than ever, e.g. workers will need to share more tasks in the future [337]. Among the top manufacturing executives, 43% think that collaboration can shorten the time to market for new products and 26% expect that improvements in terms of collaboration can reduce operational costs [338]. The necessity of better collaboration between humans and machines is highlighted by the following important statement: "Humans should never be subservient to machines and automation, but machines and automation should be subservient to humans" [339].

Therefore, future human-machine teams should be defined based on the three main features of human-machine symbiosis, namely human centrality, social wellness and adaptability [340]. The development of these balanced automation systems requires human-centred automation reference architectures to integrate the life cycle and human aspects into the Enterprise Architecture Body of Knowledge [341]. Human-machine collaborative intelligence is a partnership to optimize the benefits of teams and maximise their long-term returns with regard to interactions with the environment and other agents [340].

A part of the emerging smart factory development trend is to enhance the capabilities of human workers, where a significant part of the formalisation is the so-called smart operator (or smart worker) development field. Technologies supporting complex man-machine interactions play an essential role in improving the learning curves of operators, as presented in a study [342], which focuses on Augmented Reality and vocal interaction-based personal digital assistant solutions. As wearable devices and smart sensors become more widespread, they offer a way to integrate operators into the concept of smart factories and develop intelligent operator workspaces [343]. Further studies on Operator 4.0 and approaches for smart factory integration have also been considered. There are promising studies about cognitive solutions [344], or semantic approaches for knowledge representation, knowledge and, digital contents management within the Smart Operator domain [345].

In the next section, the methodology of higher-order network representation is described along with the possible analytics and benefits of hypergraph-based approach.

II.4.2 Higher-order network representation to support collaboration

In this section, first the modeling and mathematical tools of high-order network representation and analytical methods with hypergraphs are discussed. The background of modeling a manufacturing system is presented in Section II.4.2.1 discussing the theoretical background of hypergraph-based modeling in Section II.4.2.2. Section II.4.2.3 presents more detailed examples of hypergraph-based production processes models. Finally, the applied analytical methods of hypergraphs are outlined in Section II.4.2.4.

II.4.2.1 Hypergraphs for modeling complex manufacturing systems

The goal of high-order network representations is to identify collaboration scenarios of different actors and elements of the manufactruing process. Hypergraphs are applied for the purpose of production modeling, not only because of their efficient community structure, centrality and data clustering features but since non-pairwise interactions in the production space in the form of multiple complex relations can be described. As in the case of the network of a trivial graph, vertices and edges are also present in a hypergraph, however, the data, which is described by these network elements, can be more sophisticated. Since edges and vertices can have multiple meanings, a distinction is made between different types. This chapter only discusses undirected hypergraph networks, which means that the precedence that should be given to the process steps of the production is beyond the scope of this study.

To represent the capabilities and resources of the manufacturing, elements that are compatible with the ISA-95 standard [63, 346], the B2MML [5], or semantic representation methods of manufacturing systems were used [347, 348]. Another method that has been considered in modeling is the UML, which can be utilized to describe flexible manufacturing systems in an object-oriented way [349]. For example, the BOM of a complex product can also be represented in a semantic hypergraph-based way according to a recent study [350].

II.4.2.2 Basics of hypergraph analytics

In this section, the basic definitions and properties of hypergraphs are discussed. Furthermore, a manufacturing example for each property of the hypergraph is provided.

Formally, a **hypergraph** is a structure denoted by the incidence matrix $\mathbf{H} = \langle \mathbf{V}, \mathbf{E} \rangle$, where $\mathbf{V} = \{v_j\}_{j=1}^{n}$ denotes a set of **vertices** and $\mathbf{E} = \{e_i\}_{i=1}^{m}$ a family of **hyperedges** with each $e_i \subseteq \mathbf{V}$ [252]. In collaborative manufacturing, different types of vertices and hyperedges can be defined. Types of vertices can be, for example, resource-based such as robots and operators or defined as event-based, which can be elements of the concerning different products or steps of material handling. The hyperedges of the collaborative manufacturing model also can differ, e.g. production flow- or attribute-based hyperedges, that connect certain vertices required for a specific activity or involved in a specific capability.

In Table II.4.1, the different types of vertices in a collaborative scenario along with their characteristics are summarized. Additionally, in Table II.4.2, an overview of the possible types and the properties of the hyperedges in the collaborative manufacturing model is given. Two types of hyperedges are defined by, dividing the network into "classes" as the steps of production flow and the utilized attributes. Furthermore, some of the different network properties of these types of hyperedges are also described.

Table II.4.1: Vertex types and characteristics of the collaborative manufacturing model

	Event-based vertex	Resource-based vertex	Capability-based vertex
Properties	Probability of utilization, failure rate and takt time.	Physical characteristics, capacity and availability.	Qualitative and quantitative factors of a production element.
Aggregation	Based on similarities between properties.	Based on similarities between properties.	Based on similarities between properties.
Corresponding sub-groups	Events that happen at the same time or in a sequence.	Resources with the same usage characteristics.	Network elements related to the same resources.

Table II.4.2: Types of hyperedges and characteristics of the collaborative manufacturing model

	Production flow hyperedge	Attribute hyperedge
Definition	Represents the flow of material, energy or information within vertices.	Represents the correlation between the properties of the vertices.
Direction of the edge	The direction of the hyperloop shows the sequence of the vertices during the process step.	Directed only if the utilization of the attributes is important.
Weight of the edge	A quantitative property of the material, energy or information flow such as cost, time or quantity.	Only directed if the utilization of the attributes is important.

Hyperedges can come in different sizes $|e_i|$, ranging from singletons $\{v\} \subseteq \mathbf{V}$ (distinct from the element $v \in \mathbf{E}$) to an entire vertex set $\mathbf{V}$. Since a hyperedge $e = \{v_1, v_2\}$ where $|e| = 2$ is the same as a graph edge it follows that all graphs are hypergraphs, specifically identified as being "2-uniform" [351]. The size of a hyperedge that is, how many vertices belong to a set, includes information about the complexity of a particular step in the manufacturing process, how large a human- or machine-based workforce is required, or what type of skill configuration is necessary for a procedure.

A hypergraph $\mathbf{H}$ is determined uniquely by its Boolean **incidence matrix** $B_{n \times m}$, where $B_{j,i} = 1$ if $v_i \in e_i$ and 0 otherwise [352]. Therefore, during the modeling, the interconnecting relationships of the collaborative process in the form of a matrix can be described.

The **degree** of a vertex is the number of hyperedges to which it belongs, $d(v) = |\{e : v \in e\}|$, and the size of a hyperedge is its cardinality, $|e|$ [353]. In other words, if a vertex represents a robot in the collaboration, then the degree of the vertex shows how many work processes the robot is involved in. Furthermore, if a hyperedge represents the allocation of an operator to a workstation, then the size of that hyperedge corresponds to the importance of the process with a higher number of involved members.

Let $\mathbf{H}$ be a simple example of a manufacturing scenario, where different groups and working procedures are defined. Let $\mathbf{V}$ denote different activities a_i and operators o_j as vertices of the network. A hypergraph can be built in the following way:

- The set of vertices is the set of manpower and activities:
  $\mathbf{V} = \{o_1, o_2, a_1, a_2\}$
- The family of hyperedges $(e_i)_{i\in\{1,2,...k\}}$ is built in the following way:
  - $(e_i), i \in \{1, 2, ...k\}$ is the subset of operators or activities, which are involved in the i-th production step.

In Figure II.4.1, the example hypergraph is visualized with an Euler diagram [354], where three different hyperedges can be seen, as follows: $e_1 = \{o_1, o_2\}$, $e_2 = \{o_1, a_1\}$, $e_3 = \{a_1, a_2, o_2\}$. Furthermore, in Table II.4.3, the incidence matrix of example hypergraph $\mathbf{H}$ is shown. As an example, hyperedge e_3 represents a production step, which requires two activities, a_1 and a_2, as well as the operator o_2, who performs these activities. The same information is stored in the third line of the incidence matrix (Table II.4.3).

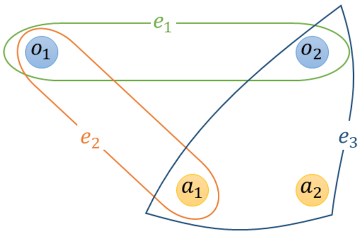


FIGURE II.4.1: The Euler diagram of the example hypergraph **H**

H	o_1	o_2	a_1	a_2
e_1	1	1	0	0
e_2	1	0	1	0
e_3	0	1	1	1

TABLE II.4.3: The incidence matrix of example hypergraph **H**

The **dual hypergraph** $\mathbf{H}^* = \langle \mathbf{V}^*, \mathbf{E}^* \rangle$ of $\mathbf{H}$ has a $\mathbf{E}^* = \{e_i^*\}_{i=1}^m$ vertex set and a family of hyperedges $\mathbf{V}^* = \{v_j^*\}_{j=1}^n$, where $v_j^* := \{e_i^* : v_j \in e_i\}$. Therefore, $\mathbf{H}^*$ is the hypergraph with the transposed incidence matrix B^T and $(\mathbf{H}^*)^* = \mathbf{H}$ [352]. Thanks to the dual hypergraph attribute, the hyperedges can be converted into vertices and vice-versa. This feature can facilitate the more in-depth structural investigation of a complex manufacturing system. For example, a hypergraph model about the investigated production system can be created, where hyperedges show the resources or actors of the process, and the vertices belong to work steps. After that, the visualization can be very quickly "inverted" to a dual form, where the hyperedges stand for the resources or actors of the system, while vertices highlight the related work steps.

In Figure II.4.2, the dual hypergraph version of the previous example is visualized, and the incidence matrix of $\mathbf{H}^*$, where $o_1 = \{e_1, e_2\}$, $o_2 = \{e_1, e_3\}$, $a_1 = \{e_2, e_3\}$

and $a_2 = \{e_3\}$, is presented in Table II.4.4. As a result, it can be said that $\mathbf{H}^*$ swaps the roles of vertices and hyperedges. Thanks to the dual graph feature, certain elements of the system can be modeled in several ways in a hypergraph-based manufacturing model. It is possible to represent a resource allocation scenario with a hyperedge or with different vertices as well.

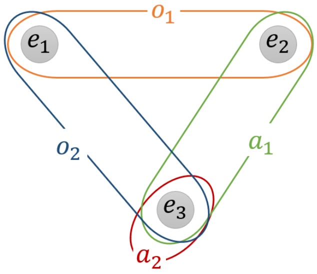


FIGURE II.4.2: The Euler diagram of the example dual hypergraph $\mathbf{H}^*$

$\mathbf{H}^*$	e_1	e_2	e_3
o_1	1	1	0
o_2	1	0	1
a_1	0	1	1
a_2	0	0	1

TABLE II.4.4: The incidence matrix of the example hypergraph $\mathbf{H}^*$

The line graph $L(H)$ of hypergraph $\mathbf{H}$ consists of a vertex set $\{e_1^*, ..., e_m^*\}$ and an edge set $\{(e_i^*, e_j^*) | e_i \cap e_j \neq \emptyset, i \neq j\}$ [355]. In order to additionally capture information about the size of intersecting hyperedges, line graphs of hypergraphs may be defined with additional edge weights, where $\{e_i^*, e_j^*\}$ has the weight $|e_i \cap e_j|$ [353].

The weight of a hyperedge ω_i is related to the frequency of occurrence of the hyperedge (multiplicity) and the cardinality of the hyperedge. There are different approaches of the hyperedge weights, namely as a constant, frequency-based, or according to the definitions by Newman, Gao or Network Theory [356]. In the case of a production environment, the weight of a hyperedge can hold information about the relevancy of a set of production members (as machines and operators in a production process step) or other cost parameters such as the training or time cost of a vertex set connected by the hyperedge.

II.4.2.3 Hypergraph-based modeling of a production process

This section aims to provide more details about how hypergraphs can be used for the modeling of the production environment.

In the case of vertices, a distinction is made between several types, such as resource-, capability- and event-based vertices, which are summarized in Table II.4.5. Resource-based vertices of the collaborative manufacturing are the human and machine members of the production process, such as operators, robots or machines. The machining-based vertices stand for manufacturing activities such as milling, drilling and material handling. Furthermore, the event-based vertices cover the production steps of manufacturing Product A or Product B.

TABLE II.4.5: Examples of different types of vertices

Resource-based vertices

Operator	$\{v_1^o, v_2^o \dots v_{N_o}^o\}$	○
Robot	$\{v_1^r, v_2^r \dots v_{N_r}^r\}$	○
Machine	$\{v_1^m, v_2^m \dots v_{N_m}^m\}$	○

Machine-based vertices

Milling	$\{v_1^{mi}, v_2^{mi} \dots v_{N_{mi}}^{mi}\}$	□
Drilling	$\{v_1^d, v_2^d \dots v_{N_d}^d\}$	□
Material handling	$\{v_1^{ha}, v_2^{ha} \dots v_{N_{ha}}^{ha}\}$	□

Event-based vertices

Production of A	$\{v_1^{pa}, v_2^{pa} \dots v_{N_{pa}}^{pa}\}$	△
Production of B	$\{v_1^{pb}, v_2^{pb} \dots v_{N_{pb}}^{pb}\}$	△

In the proposed example model, activity and attribute-based edges are included as listed in Table II.4.6. An activity-based hyperedge connects the vertices involved in a specific production procedure or can be defined as a set of vertices which represent collaborating resources that perform an activity. The weight of an edge can equate to the time or cost of the whole activity. Furthermore, the weight of the same hyperedge can differ within vertices which are connected to it. The other type of hyperedge in the proposed modeling methodology is attribute-based, which connects vertices with a specific type of attribute or characteristic and the weight of these edges determines the suitability.

It is important to mention that in a hypergraph network, some of the vertices and edges are convertible such as capability which can occur as an edge or as a vertex.

TABLE II.4.6: Examples of different types of hyperedges

Hyperedges		
Activity-based	$\{e_1^a,\ e_2^a \ldots\ e_{N_a}^a\}$	│
Attribute-based /capability/	$\{e_1^c,\ e_2^c \ldots\ e_{N_c}^c\}$	‖

Moreover, a further generalization in a hypergraph is that a hyperedge as well may not only contain vertices but other hyperedges [252].

In Figure II.4.3-**a**, a hypergraph representation example is visualized, where there are two different types of hyperedges and three types of resource-based vertices. In Table II.4.7, the incidence matrix of hypergraph $\mathbf{H}_1$ is listed from Figure II.4.3-**a**. To accomplish the activity covered by hyperedge e_1^a, the following vertices need to be involved: v_1^o, v_2^o and v_1^m, so two operators and one machine. Another hyperedge referred to as e_1^c connects vertices v_1^r and v_2^o and acts as an attribute-based hyperedge, which connects a *robot* and an *operator*-type, resource-based vertex.

In Figure II.4.3-**b**, a bit more complex hypergraph representation of a manufacturing process is visualized. To accomplish the activity covered by hyperedge e_4^a, the following vertices need to be involved: v_1^{Pa}, v_2^{Pa}, v_1^{mi} and v_1^d. Therefore, *Product*

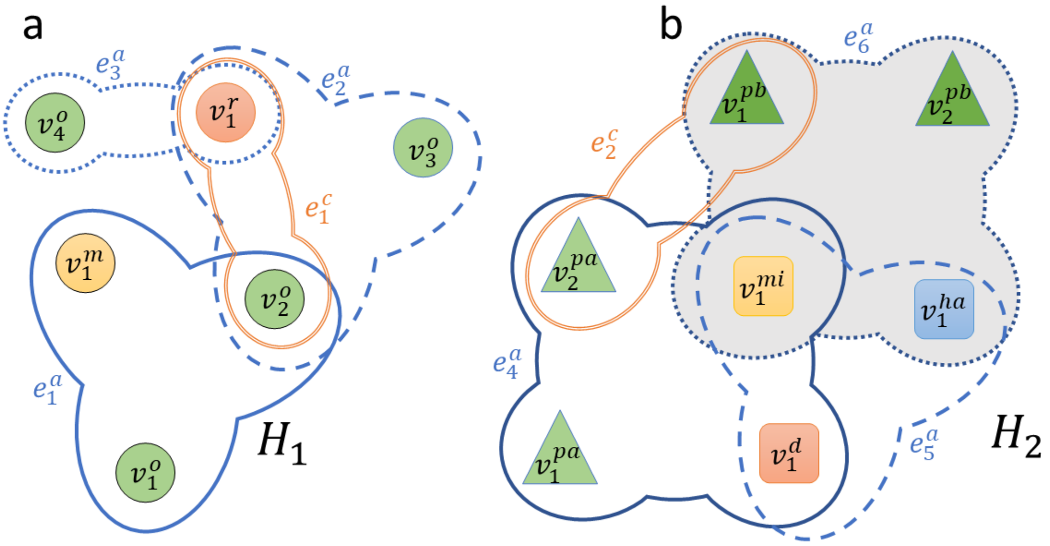


FIGURE II.4.3: $\mathbf{H}_1 = \langle \mathbf{V}_1; \mathbf{E}_1 \rangle$ is a hypergraph representation of resource-based vertices allocated by different types of hyperedges where $\mathbf{V}_1 = \{v_1^o, v_2^o, v_3^o, v_4^o, v_1^m, v_1^r\}$ and $\mathbf{E}_1 = \{e_1^a, e_2^a, e_3^a, e_1^c\}$ moreover, $\mathbf{H}_2 = \langle \mathbf{V}_2; \mathbf{E}_2 \rangle$ is a hypergraph representation of machine and event-based vertices allocated by different types of hyperedges where $\mathbf{V}_2 = \{v_1^{pa}, v_2^{pa}, v_1^{mi}, v_1^d, v_1^{ha}, v_1^{pb}, v_2^{pb}\}$ and $\mathbf{E}_2 = \{e_4^a, e_5^a, e_6^a, e_2^c\}$

Table II.4.7: The incidence matrix of example hypergraph $\mathbf{H}_1$

	v_1^o	v_2^o	v_3^o	v_4^o	v_1^m	v_1^r
e_1^a	1	1	0	0	1	0
e_2^a	0	1	1	0	0	1
e_3^a	0	0	0	1	0	1
e_1^c	0	1	0	0	0	1

A-type, event-based, and two other machining-based vertices are found in this set. However, in the next set of vertex connected by activity-based hyperedge e_5^a three different types of machining-based vertices are covered, namely v_1^{mi}, v_1^{ha} and v_1^d. Finally, hyperedge e_6^a contains two types of *Product B* , that is event-based vertices (v_1^{Pb} and v_2^{Pb}) and two other machining-based vertices (v_1^{mi} and v_1^{ha}). An attribute-based hyperedge on this visualization is also present, where e_2^c connects the vertices v_2^{Pa} and v_1^{Pb}, providing an example where hypergraphs, edges and vertices can deliver the same information as all three can describe attributes here.

After describing the principles and modeling examples of hypergraphs, more complex network analytical methods are discussed in the following subsection.

II.4.2.4 Advanced hypergraph-based analysis of a collaborative manufacturing

This subsection aims to provide network-based metrics that are suggested for hypergraph-based analysis of the collaborative manufacturing model. In Table II.4.8, the studied hypergraph-specific centrality and analytical methods were summarized and examples of application scenarios in manufacturing analytics given.

In order to define the hypergraph centrality measures, first, the notions of a hypergraph walk and distance are introduced [353]. Given two hyperedges $e, f \in E$, an s-walk of length k between e and f is a sequence of hyperedges $e_0, e_1, ..., e_k$ such that $e_0 = e, e_k = f$ and $s \leq |e_i \cap e_{i+1}|$ for all $0 \leq i \leq k-1$. In other words, an

Table II.4.8: Types of hypernetwork measures and their application in collaborative manufacturing model analysis

Centrality metric	Application in collaborative manufacturing
S-betweenness centrality	Show the importance of the elements
S-closeness centrality	Show how an element is shared
Modularity of the hypergraph	Show how modular the production process is

s-walk is a sequence of edges such that the size of pairwise intersections between neighboring edges is at least s.

The s-distance, for a fixed $s > 0$ is defined as $d_s(e, f)$ between two edges $e, f \in E$, as the length of the shortest s-walk between them. If no s-walk is found between two edges, then the s-distance is infinite [353]. Two edges $e, f \in E$ are defined as s-adjacent if $|e \cap f| \geq s$ for $s \geq 1$ [357]. In addition, the s-diameter is defined as the maximum s-distance between any two edges and the s-component as a set of edges connected pairwise by an s-walk [352]. The s-path is referred to in the case of s-walks, where the edges are not repeated, so any hyperedges from the hypergraph can participate only once in the path.

Furthermore, the walks in hypergraphs also have a certain width. In Figure II.4.4, three examples of walks in hypergraphs (based on Ref. [352]) are shown. In the first simple example (a), a 2-uniform hypergraph is presented. The length of the walk between hyperedges e_1 and e_3 is two (as the walk needs to go through e_1 and e_2 to reach e_3), and its width is one (as the number of vertices at interconnecting hyperedges is one). In the second scenario (b), a hypergraph is presented where the length between e_1 and e_3 is two and the interaction is one. While in the third case (c), its length is still two, but the interactions are stronger as its width is three (because the minimum number of vertices at interconnecting hyperedges is three).

Aksoy *et al.* [353] defined several network science methods generalized from graphs to hypergraphs, including vertex degrees, diameters and clustering coefficients. In this paper, their generalization of the betweenness centrality and closeness centrality with regard to hypergraphs is applied using the stratification parameter s.

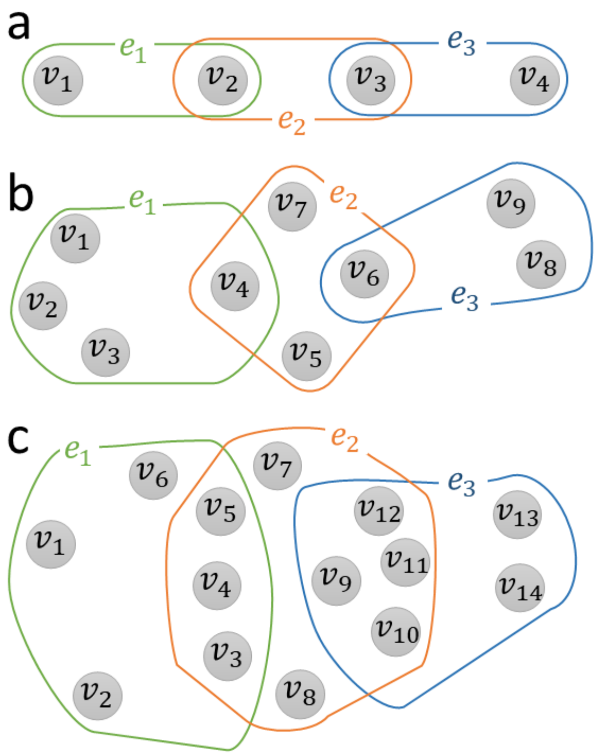


FIGURE II.4.4: Examples of walks in hypergraphs: (a) a 2-uniform hypergraph of length two and width one between e_1 and e_3, (b) one of length two and width one between e_1 and e_3, and (c) another of length of two and width three between e_1 and e_3

The s-betweenness centrality of edge e is:

$$BC_s(e) := \sum_{f \neq e \neq g \in E} \frac{\sigma^s_{fg}(e)}{\sigma^s_{fg}}, \tag{II.4.1}$$

where σ^s_{fg} denotes the total number of the shortest s-walks from edge f to edge g and $\sigma^s_{fg}(e)$ represents the number of those shortest s-walks that contain edge e [351].

The harmonic s-closeness centrality of an edge e is the reciprocal of the harmonic mean of all distances from e:

$$HCC_s(e) := \frac{1}{|E_s| - 1} \sum_{f \in E_s, f \neq e} \frac{1}{d_s(e, f)}, \tag{II.4.2}$$

where $E_s = \{e \in E : |e| \geq s\}$ [351].

In order to take into account multiple s values simultaneously in the analysis, the average of the centrality values across a range of s values is calculated and the average s-betweenness centrality [351] is defined as:

$$\overline{\mathrm{BC}}_s(e) = \frac{1}{s} \sum_{i=1}^{s} BC_i(e), \tag{II.4.3}$$

and the average harmonic s-closeness centrality [351] as:

$$\overline{\mathrm{HCC}}_s(e) = \frac{1}{s} \sum_{i=1}^{s} HCC_i(e). \tag{II.4.4}$$

An example visualization is presented in Figure II.4.5 to demonstrate the behavior of centrality metrics in hypergraphs. The s-closeness centrality can be represented by a hyperedge (or hyperedges), which can most easily reach all other hyperedges in the hypergraph. In Figure II.4.5, the high closeness centrality element is highlighted in purple as the average distances from edges i, g, k and h are minimal compared to other groups. The s-betweenness centrality provides information about which hyperedge (or hyperedges) has the most control over the flow between other hyperedges and groups. In Figure II.4.5, the high betweenness centrality element is denoted in brown, as the maximum number of shortest paths go from edges k and m since they bridge two parts of the network.

Furthermore, in Figure II.4.5, two examples of s-walks on this slightly more complex hypergraph are presented. W_{a-e} s-walk on the left-hand side has a length of four between a and e as well as a width of one, and the s-component is 19 as W_{a-e} contains 19 vertices. W_{m-o} s-walk on the right-hand side is of length one between o and m as well as has a width of two, and the s-component is five as W_{m-o} contains five vertices.

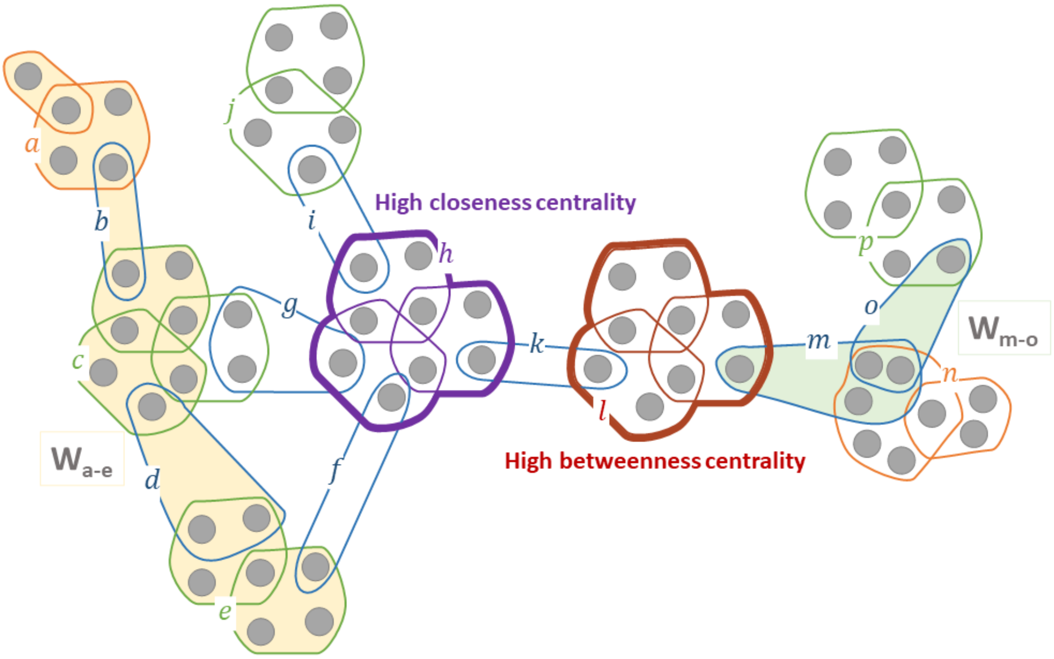


Figure II.4.5: Example hypergraph to demonstrate closeness and betweenness centrality metrics and s-walks

Based on the previous discussion and Table II.4.8, it can be concluded that hypergraph-based centrality metrics can be utilized in the design of collaborative manufacturing, if:

- a job competency has a high degree of centrality, then it is highly utilized and critical in the production process;
- an operator has a high degree of centrality, then the scheduling of his/her work is critical (assembly line balancing);
- the path between two elements (as work areas) is relatively large, then it can be decomposed and the procedure reallocated.

II.4.3 Designing collaborative manufacturing for a wire harness assembly process

This section presents how the proposed method can be applied to the analysis and redesign of a manufacturing system. The case study of this chapter also comes from the field of the wire harness assembly industry. Based on the production processes, the case study is motivated by a multinational wire harness factory. However, due to confidentiality policies, detailed information cannot be published, but the validation of the proposed methodology is continuous with the production experts.

To present the hypergraph-based methodology, a more complex, wire harness assembly-based benchmark problem has been applied, which is described in Appendix C.2 of Chapter C. A small production line with batch and conventional production was chosen and Figure C.4 of the Appendix shows the process flow, which is based on a real assembly line. The process contains two assembly lines that produce shared tasks and resources in parallel.

In Section II.4.3.1, several visualization applications with hypergraphs are shown. Section II.4.3.2 describes the analytical method to identify critical elements and collaborations of the manufacturing process. In Section II.4.3.3, the segmentation process is presented. Finally, Section II.4.3.4 discusses the benefits of the proposed hypergraph-based methodology.

II.4.3.1 Hypergraph-based representation of collaborative manufacturing designed for the wire harness assembly line

In this subsection, the designed collaborative manufacturing is presented by visualizing the hypergraph model in three different ways, which correspond exactly to the data, however, different valuable conclusions can be drawn from them.

First, the serialized incidence matrix of the hypergraph is presented (Figure II.4.6), then the normal- (Figure II.4.7) and dual- (Figure II.4.8) hypergraphs of the wire harness assembly process are shown. In Figure II.4.6, the serialized incidence matrix of the wire harness manufacturing-based case study is visualized. On the vertical axes, the 70 different activities are listed having been re-ordered, while on the horizontal axes, the human-machine resources, capabilities, and other tooling or sensor elements of the collaborative manufacturing example are provided.

In Figure II.4.6, after serialization of biadjacency matrix (**B**) and identifying clusters in the data. The closely connected activities and items of the collaborative manufacturing are highlighted. The top bicluster, highlighted in yellow, is denoted by a vertex set, namely capability v^c_7, machine v^m_1, and operator v^o_2, in the case of the following activities: e^a_5, e^a_6, e^a_7, e^a_8 and e^a_9. Since activities e^a_6 and e^a_7 from this group also connect with the v^s_3 sensor, these five activity-based hyperedges create a bicluster in the collaborative manufacturing model because the same operator, machine and capability are utilized in these production steps. The second bicluster, denoted in purple, consists of sensor v^s_1, AGV v_a and the capability v^c_8 in the case of the following activity-based hyperedges: e^a_{17}, e^a_{33}, e^a_{38} and e^a_{19} as ll as e^a_1, e^a_{35}, e^a_{36} and e^a_{70}. Another bicluster denoted in red is related to AGV v_a and sensorv^s_2, in the case of activity-based hyperedges e^a_2, e^a_{18}, e^a_{34}, e^a_{37}, e^a_{53} and e^a_{69}. Furthermore, more possible clusters are highlighted with dashed lines in Figure II.4.6.

In Figure II.4.7, the normal hypergraph of the production network is visualized, which shows how activities involve the elements of collaborative manufacturing. This representation helps to identify what central elements affect the activities most, e.g. activity-based hyperedge e^a_{31} connects vertices v^d_2, v^s_3, v^r_3, v^o_4 and v^c_6.

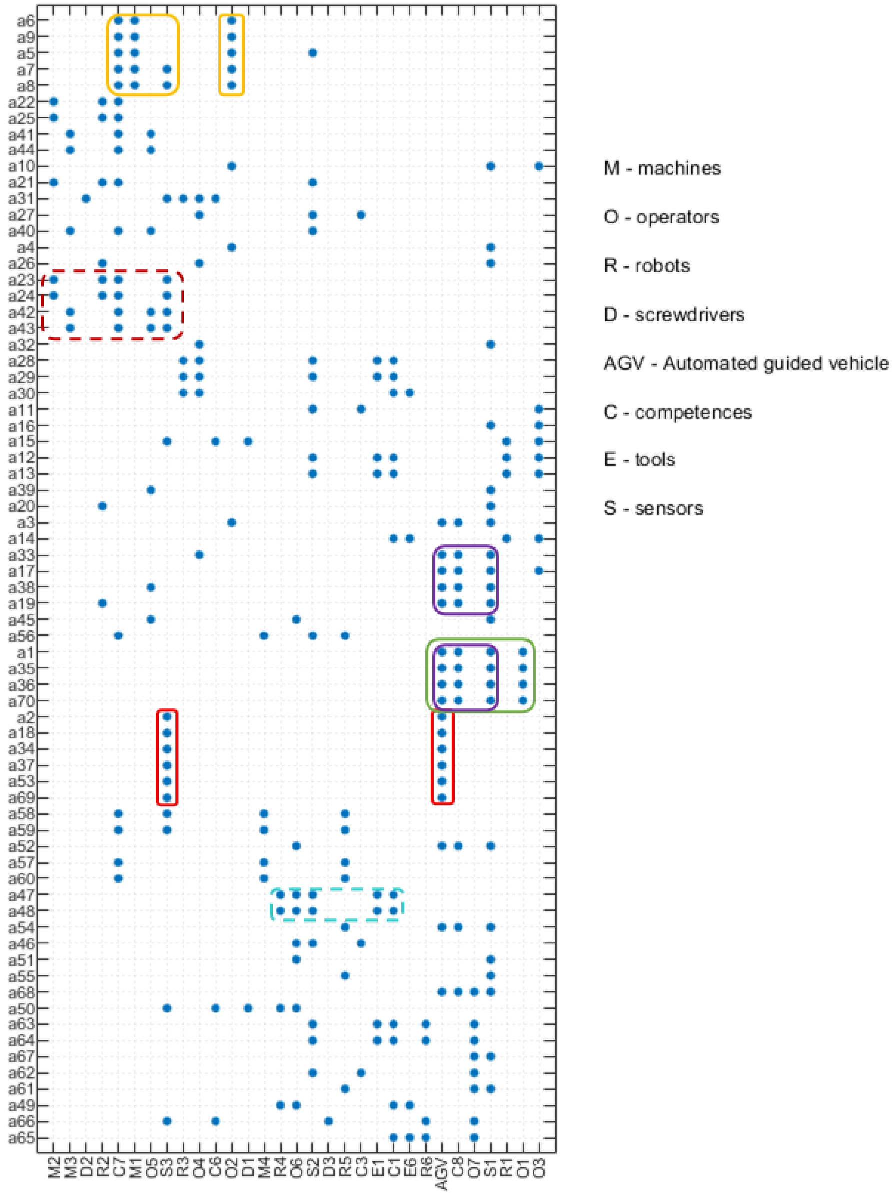


FIGURE II.4.6: The serialized incidence matrix of the collaborative manufacturing concerning wire harness manufacturing. Some biclusters of the closely connected activities and items are also highlighted.

Figure II.4.8 shows a dual hypergraph with opposite meaning to the previous figure, as it represents the assets and workers with regard to the activities involved, transposed form of the previous visualization. For example capability-based hyperedge e_3^c connects the following activities in the form of vertices: v_{11}^a, v_{27}^a, v_{46}^a and v_{62}^a. Alternatively, in a similar way, it can be sees that screwdriver $D1$-based hyperedge e_1^d is denoted by activities v_{15}^a and v_{50}^a. In this representation, hyperedge e_1^s is the largest as it contains 24 different activity-based vertices, while e_1^d and e_2^d (screwdriver-based hyperedges) are the simplest ones with only one vertex each.

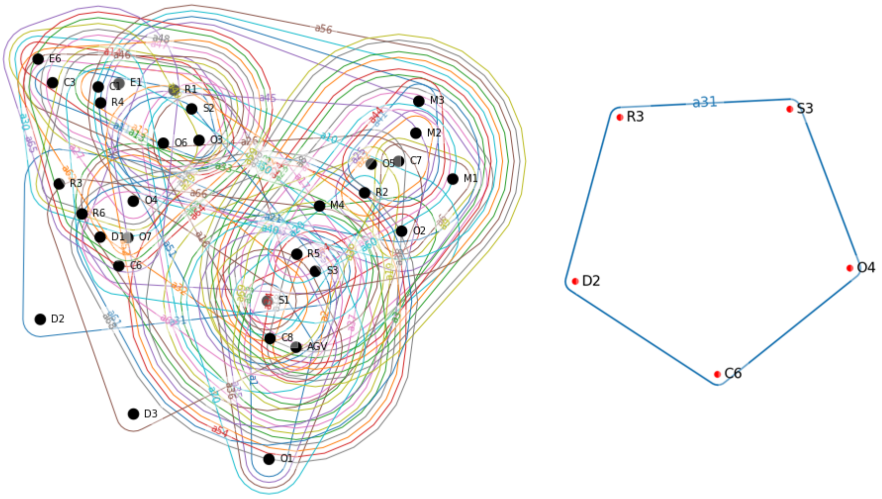


FIGURE II.4.7: Hypergraph visualization of the wire harness benchmark (on the left-hand side) - The activities become involved as a result of the elements of collaborative manufacturing and activity-based hyperedge a_{31} (on the right-hand side)

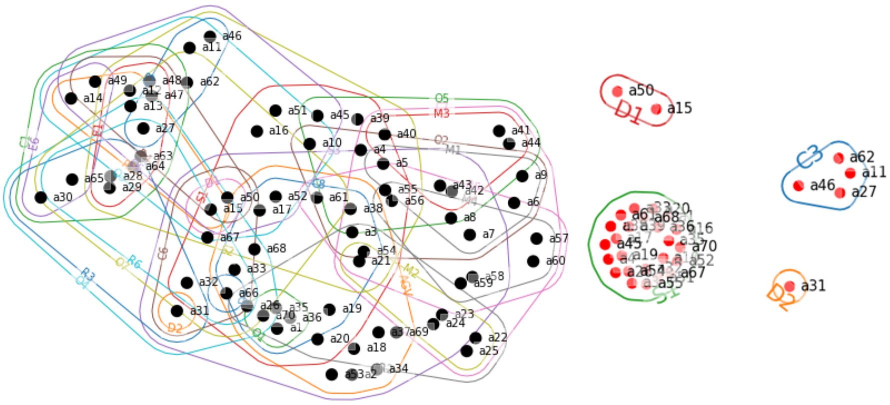


FIGURE II.4.8: Dual hypergraph visualization of the wire harness benchmark (on the left-hand side) - The elements of collaborative manufacturing become involved as a result of the activities and hyperedges $D1$, $S1$, $C3$ and $D2$

II.4.3.2 Identification of the critical elements and collaborations

The critical elements and collaboration scenarios of the collaborative manufacturing can be identified based on the s-closeness and s-betweenness measures presented in Section II.4.2.4. The closeness centrality measure indicates how close a vertex is to all other vertices in the network, while the betweenness centrality detects the degree of influence a vertex has over the flow of information in the hypergraph.

Based on the s-closeness and s-betweenness metrics, the most important elements are v_2^s (sensor $S2$ as a vertex - RTLS (Real-time locating system)) and v_3^s (sensor $S3$ as a vertex - machine log) as the s-closeness values are 0.77 and 0.75 and the s-betweenness ones are 55.68 and 45.57, respectively. The results show that the most important elements (as central elements are present) of the modeled process are the sensors, given that they are connected to the most activities. The central operators are v_3^o, v_4^o, v_6^o and v_7^o with values of s-closeness and s-betweenness of 0.6 and 16, respectively. A similar conclusion can be reached if the dual hypergraph in Figure II.4.8 is investigated, where the central elements are determinative cooperation or interaction. The four operators mentioned above are found in the same vertices to the left of the center of the figure and several overlaps can be noticed.

The average s-closeness centrality value of the wire harness benchmark network is 0.598, which means the vertices have a higher probability of being closer to each other in a network than far apart. Additionally, the average s-betweenness of the hypergraph network is 9.393, although it has a high deviation because most of the vertices have a high influence on the hypergraph.

Based on the determinative hyperedges, in the case of closeness and betweenness, the same activity type was the most significant, that is t_{19} (Positioning of a crimp into a vise), which is usually handled by two operators or one robot, while applying capability $C7$ and monitoring the process using sensor $S2$ (RTLS). The related activities to activity type t_{19} are v_5^a, v_{21}^a, v_{40}^a and v_{56}^a, which are visualized in Figure II.4.9 with a dual hypergraph.

The representation of the hypergraph can show the central element of the complex system based on the higher-order network representation. In the following subsection, how these higher order connections can be investigated with the s-walk analysis will be shown and be used for segmentation tasks such as forming manufacturing cells.

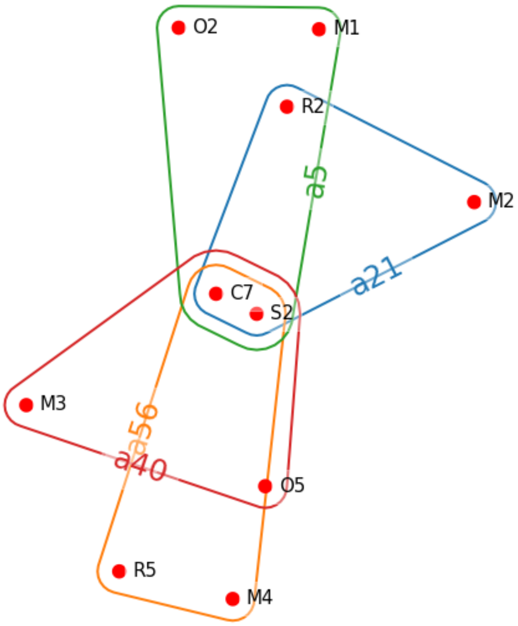


FIGURE II.4.9: Dual hypergraph representation of activity type t_{19} related assembly activities (hyperedges), and the resources, actors are shown as vertices

II.4.3.3 Segmentation of the collaborative manufacturing model

The centrality metrics presented in the previous subsection facilitate the detection of the critical or potential collaboration areas in the manufacturing network. While this subsection describes in detail how the system can be segmented, which helps to investigate them more in-depth. The hypergraph representation provides information to determine the strongly interdependent elements as the modules can be identified. These modules are the bases for forming manufacturing cells, since they show what elements should be planned together and how the process can be decomposed. The first task is to determine the strongly connected elements.

The s-walk methods are applicable to measure the connectedness in the collaborative processes where multiple participants exist. The benefit of the hypergraph representation is that the second and third walks between the vertices where these walks represent closely connected elements can also be seen. For example, the s-betweenness value in the case of v_1^s (sensor $S1$ as a vertex (*camera*) - highlighted in orange in the figure) is significantly higher when the second walk is calculated, as it rises from 13.99 (7^{th} place) to 102.53 (2^{nd} place). Furthermore, the betweenness value of v_7^c increases by more than ten times, as it is a required capability ($C7$ - highlighted in blue in the figure) for the crimping step. The graph representation of the connections between the elements in the case of the first and second s-walks can be seen in Figure II.4.10. Based on the resulting graph of the second s-walk and the s-betweenness value of two (as described below), it can be noticed that the centrality of $S1$ and $C7$ with regard to betweenness is described by the influence of the vertex.

The crimping competency ($C7$) is highly relevant as the production flow has four crimping stations with several shared resources. Logically, $S1$, that is, the camera sensor used to monitor many collaborative tasks such as AGV loads and transport between operators, should be given a high level of importance. In Figure II.4.8, it can be noticed that $S1$ covers a lot of activities. This vertex ($S1$) is denoted by the red line in the middle of Figure II.4.8 which covers many activities and is connected to several other elements.

The significant connections can be determined by modularity analyses. Several algorithms are used to identify modules in a network. The Louvain algorithm was applied to find some communities based on the activities and elements (human workers, robots, etc.). The algorithm identifies five activity-based communities:

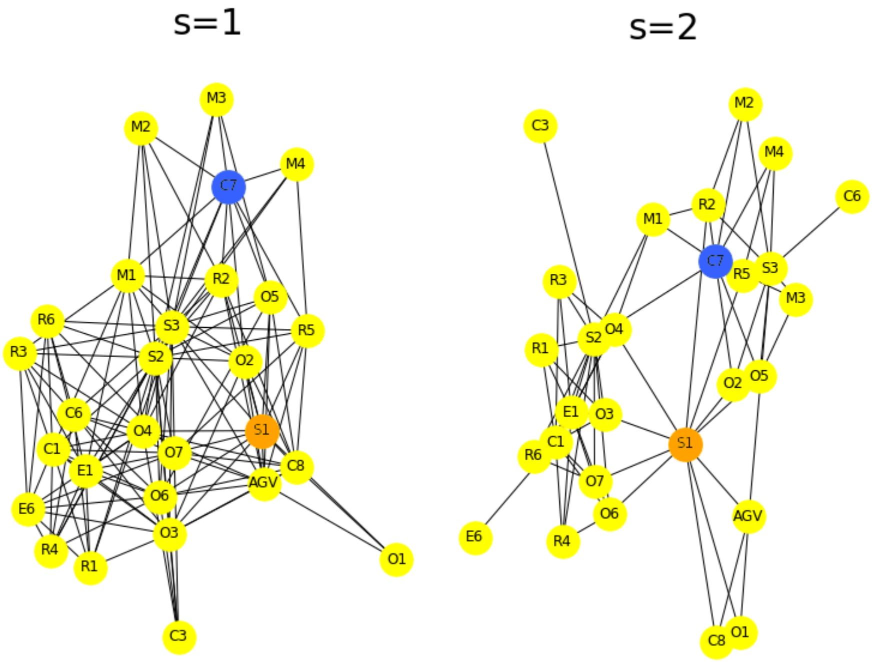


FIGURE II.4.10: The $s = 1$ and $s = 2$ walks highlight the connectedness of the elements of the designed collaborative manufacturing model

1. *Crimping* 3 - related activities
2. *Assembly* $1-4$ - related activities
3. AGV - related activities
4. *Crimping* $1-2$ - related activities
5. *Crimping* 4 - related activities

The Louvain algorithm is applied to the dual hypergraph to analyse the main elements of the collaborative manufacturing model. Three modules were identified from the elements, the first contains all AGV-related vertices such as the AGV, the loading of the AGV capability and operator $O1$ as they only work together with the AGV. Since this module also includes the camera and machine-log sensor, it determines the elements that are collaborative at multiple stations. The crimping machine-related elements are found in the second module, e.g. robots, crimping machines, related operators and capability $C7$. The third module consists of the elements related to the assembly stations with the RTLS sensor.

The central element is the key to collaboration, and this result shows what is the most critical. The central elements are the s-walk method, which provides valuable information about the complex collaborative processes, where multiple resources work together and cooperate with each other. In this case, the crimping capability is significant (in the case of the second s-walk), which shows us that training more and more operators to use the crimping station together with robots should be considered. The modules help to discover the joint elements and divide the complex problem into the most significant parts. The results identify the three significant parts of the investigated use case.

In Figure II.4.11, a part of the wire harness assembly-based case study is visualized. At the top of the figure, the hypergraph network is presented with activity-based hyperedges, where robot $R2$, operator $O3$, robot $R1$, and the AGV are chosen as key elements based on centrality metrics. The four elements are visualized at the bottom along with all the other related activities as vertices in the dual hypergraph to demonstrate the benefit of dual hypergraph representations. This approach could give further information about the other related assembly activities with the dual graph form after determining the four central elements. Furthermore, on the bottom dual hypergraph visualization, activities v^a_{17} and v^a_{19} as interconnecting

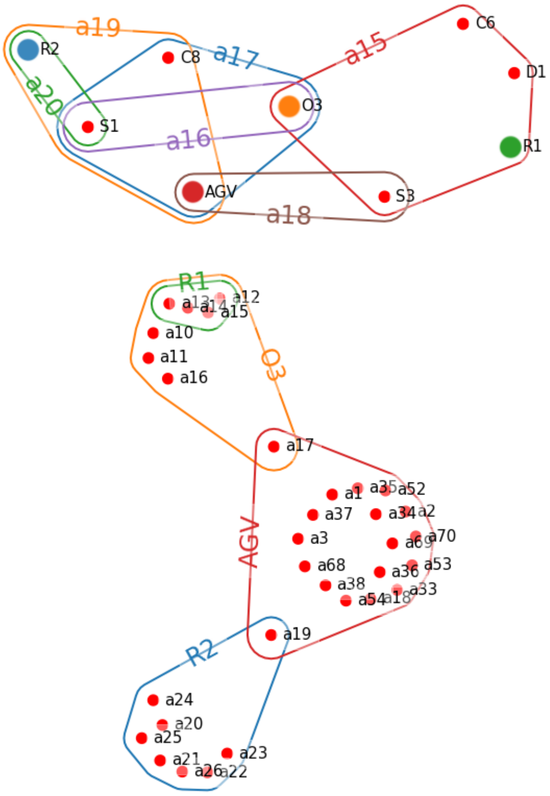


FIGURE II.4.11: Hypergraph (at the top) and dual hypergraph (at the bottom) representations of a collaboration scenario, where operator $O3$, robots $R1$-$R2$ and the AGV are the focal points

steps within the AGV and operator $O3$ or robot $R2$ can also be seen. An example of collaboration is the overlapping section of robot $R1$ and operator $O3$ on the bottom dual graph representation, where v^a_{12}, v^a_{13}, v^a_{14} and v^a_{15} vertices belong to activities.

Figure II.4.12 aims to demonstrate the features of the hypergraph-based visualization of the collaboration analysis of a manufacturing system. Robots, operators, and the AGV are visualized in the form of hyperedges, while the red vertices belong to assembly activities. Within the AGV hyperedge in Figure II.4.12 the activity vertices (red dots) overlapped with other hyperedges show scenarios when operator or robot actors work together at the same time and "share" activities. Collaboration cases are also highlighted on the hypergraph, such as operator $O7$ collaborating with robot $R6$ and having a shared activity with robot $R5$ and with the AGV. In a more complex, real industrial environment, the proposed method can also facilitate the detection of critical zones, scheduling processes, improve ergonomic aspects during collaboration or layout design.

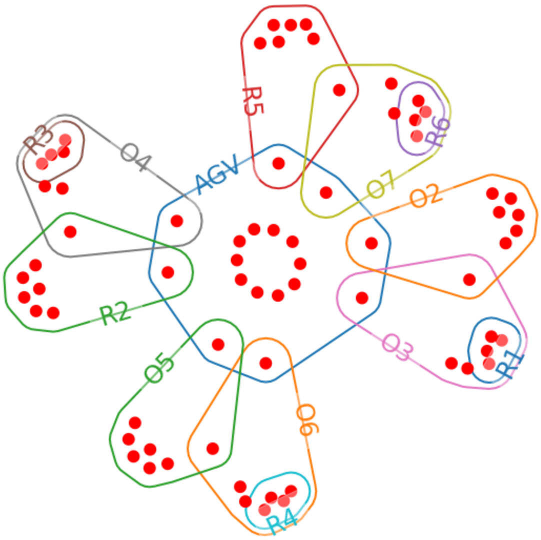


Figure II.4.12: Dual hypergraph representation of the collaboration between Operators, Robots, and the AGV, where the red vertices belong to different activities

II.4.3.4 Discussion on the benefits of the hypergraph-based analysis and suggestions for future research

The wire harness assembly case study-based examples presented in the previous subsection highlighted how an existing production system could be analysed as a hypergraph network. Compared to classical and advanced multilayer network-based analysis [216], the main benefit of hypergraphs is that it allows the set-theory-based analysis of the system. Sets represented by hyperedges can be used to study redundancy and resilience, and the intersection of sets can explore the flexibility of configurations. Higher-order network representations can better represent the superstructures of complex manufacturing systems where the superstructure is constructed from a set of alternatives.

A critical aspect of the research is the technologies needed to utilize the proposed concept in a real-world industrial application. Complex manufacturing system representation and analytics require a comprehensive data management system that covers all aspects of production. Information management of future manufacturing requires an effective solution as knowledge graphs or knowledge hypergraphs [358]. These solutions use a graph-based data model to capture knowledge in application scenarios that involve integrating, managing and extracting value from diverse data sources, even at a large scale [359]. Knowledge graph methods can mine information from structured, semi-structured, or even unstructured data sources and finally integrate the information into knowledge, represented in a graph [360]. Enabling technologies for the proposed hypergraph-based approach are the adequate MES and MOM supported by semantic technologies, such as ontologies and knowledge graphs [94].

An essential additional question is what can be done with the analysis results and how the uncovered knowledge can be applied to improve the manufacturing process. It has been demonstrated that the wide range of hypergraph-based metrics provides much more possibilities than classical network centralities, mainly when collaboration should be analysed. Collaborating actors whit a high influence on the production are detectable with s-walks, and the central collaborators of the intelligent manufacturing network can be found with s-closeness and s-betweenness metrics.

A non-applied but beneficial analysis tool of hypergraphs is the so-called vertex simplification, which can be used to redesign the systems by exploring the bottlenecks and critical elements of the collaborations. A method for this is a (weighted) clique expansion performed on the line graph of the dual of a hypergraph generated based on the similarities between vertices [355].

A further advantageous feature worth studying in the future is the utilization of fuzzy set memberships in fuzzy hypergraphs [361]. A fuzzy representation of a collaborative process makes it possible to store even more detailed information in the model, such as the availability or the effectiveness of allocating an operator or activity. Such representation would allow to calculate the total of the rows of the fuzzy incidence matrix. Additionally, the total FTE (Full-Time Equivalent) of the allocated operators can be obtained by summarizing the weights of vertices. A so-called Fuzzy Competition Hypergraphs method [362] can facilitate decision making, which could also be adaptable in an intelligent manufacturing environment.

II.4.4 Summary of hypergraph-based analysis of collaborative manufacturing

This work investigated the support of human-machine and human-human cooperation in manufacturing. Based on the simultaneous and integrated monitoring of the activities of the machines, robots, operators and mobile robots, additional functions that facilitate cooperation can be developed. The analysis and design of collaborative manufacturing require a tool that provides information about the impacts of their interactions.

This work highlighted that hypergraphs could support the analysis and design of manufacturing systems. The vertices of the hypergraph can represent events, resources/assets or capabilities, while the hyperedges represent the sets formed according to the activities/cooperations or attribute-type relationships. The hypergraph centrality measures and clusters/modules of the resultant network highlight the critical elements and interactions.

When necessary, the highlighted weakly connected components could be integrated by redesigning the system. The model also supports the analysis of the robustness of the manufacturing. As it is unclear what kind of simulated perturbations should be studied and which network measures should be analysed for this purpose, developing the proposed method for business process redesign could be the main research topic in this new field.

Chapter B

Conclusions and thesis findings

The previous chapters discussed the theoretical and practical results of my doctoral program. The present chapter aims to summarize the contributions made to the research of ontology-based development of Industry 4.0 & 5.0 solutions.

The motivation behind the work was pointed out in Chapter A, which are the horizontal and vertical integration in the industry while aiming for interoperability and standardization, and improved data access in ERP and MES systems. Additional goals were to develop efficient analysis and optimization methods using pre-structured, graph-based data and focus on Industry 4.0 aspects, especially Industry 5.0, where the support of collaboration and shop floor workers are aimed at the human-centric approach. Based on the problem statement, a framework for ontology-based development of Industry 4.0 & 5.0 solutions has been proposed. I highlighted, that semantic technologies make it possible to adapt network-based process models to industry standards and contextualize process data with graph-based representation, enabling interoperability and re-usability factors and making accessible a variety of process analysis and optimization methods.

As the motivation of this work consisted of modeling and optimization tasks as well, therefore the chapters are divided into two parts. First, in Part I., the semantic-based modeling, using ontologies and knowledge graphs, was presented, followed by two detailed application examples. Then in Part II., the network science-based process optimization was discussed, advanced manufacturing analytics was applied using graph-based data access, and three different methods were presented.

As the main introduction of this work in Chapter A, I investigated the main related research topics of industrial application of semantic technologies, such as the standards and ontology-based modeling of manufacturing, the field of production models, and the human-centric and collaborative approach as main challenges of Industry 5.0. After that, I made the problem statement and proposed an ontology-based framework to solve the formulated problems.

Starting Part I., first, in Chapter I.1, I presented a systematic overview of ontologies that can be utilized in building Industry 4.0 applications and highlighted ontologies that are suitable for manufacturing management. Additionally, I also discussed the industry-related standards and other related models. At the end of the introduction to the semantic technologies chapter, I summarized the main benefits and general application examples of semantic technologies, such as model digital twins, data mining, root cause analysis, or performing intelligent resource allocation.

In chapter I.2, I presented a detailed ontology-based modeling method in a wire harness manufacturing-based case study. Starting with the applied software tools of ontology-based modeling, I guided the reader through each step of development as the creation of a production-based knowledge graph, the data queries, and the evaluation of ontology data.

After that, in Chapter I.3, I discussed a more complex, knowledge graph-based framework to support human-centered collaborative and ergonomic manufacturing in Industry 5.0. First, I introduced the Human-centered knowledge graph design concept in detail and then demonstrated its application in an industrial case study. In this chapter, I investigated the simultaneous and supportive human-robot collaboration scenarios and related performance indicators. I presented the development steps of the human-centric knowledge graph, using the specific use case data and several graph-based analytics, such as robot allocation or capability analysis. Additionally, I highlighted in the results that the developed knowledge graph is capable of detecting different types of collaboration between human and machine actors in the assembly process.

In Part II. of the doctoral thesis, first in Chapter II.1, I highlighted the problem statement and introduced the theoretical and research background of network science-based process optimization, such as graph-based analytics, assembly line balancing or community detection.

As the first optimization application, in Chapter II.2, I presented a method to solve assembly line balancing with the combination of analytic hierarchy process and multilayer network-based modeling. I demonstrated the simulated annealing algorithm-based method with a complex, multilayer analysis of a wire-harness assembly graph network, where I aimed to perform multi-objective optimization.

With the second optimization algorithm in Chapter II.3, I aimed to create a modularity-based network community detection method integrating crossing minimization and bottom-up segmentation. The presented method can perform scalable and time-efficient community detection compared with other Louvain-based procedures. Thanks to the pre-serialization of the adjacency matrix, the algorithm reduced the iteration cost, as it does not need to test the entire data set. I highlighted that integrating barycentric serialization with modularity maximisation based bottom-up segmentation offers an efficient solution, especially on large data sets. I proved the efficiency of the developed method on benchmark problems using real-life and generated networks.

Finally, in Chapter II.4, I proposed a hypergraph-based analysis method to investigate collaborative manufacturing processes. I utilized the concept of intelligent space, which supports the design of human-machine and human-human cooperation in manufacturing. I also demonstrated the efficiency of hypergraph-based analysis on a collaboration-related industrial case study. I highlighted that the hypergraph centrality measures and clusters/modules of the resultant network could show the critical elements and interactions. Additionally, the vertices of the hypergraph can represent events, resources/assets or capabilities, while the hyperedges represent the sets formed according to the activities, cooperations or attribute-type relationships.

As a graphical summary, Figure B.1 represents the above-discussed theoretical and practical sections of this work. The blue-colored line stands for the two modeling applications of chapters I.2-I.3, which are the first two findings. While the orange-colored optimization line is related to the network science-based applications of chapters II.2-II.4 and forms the three additional findings of the thesis.

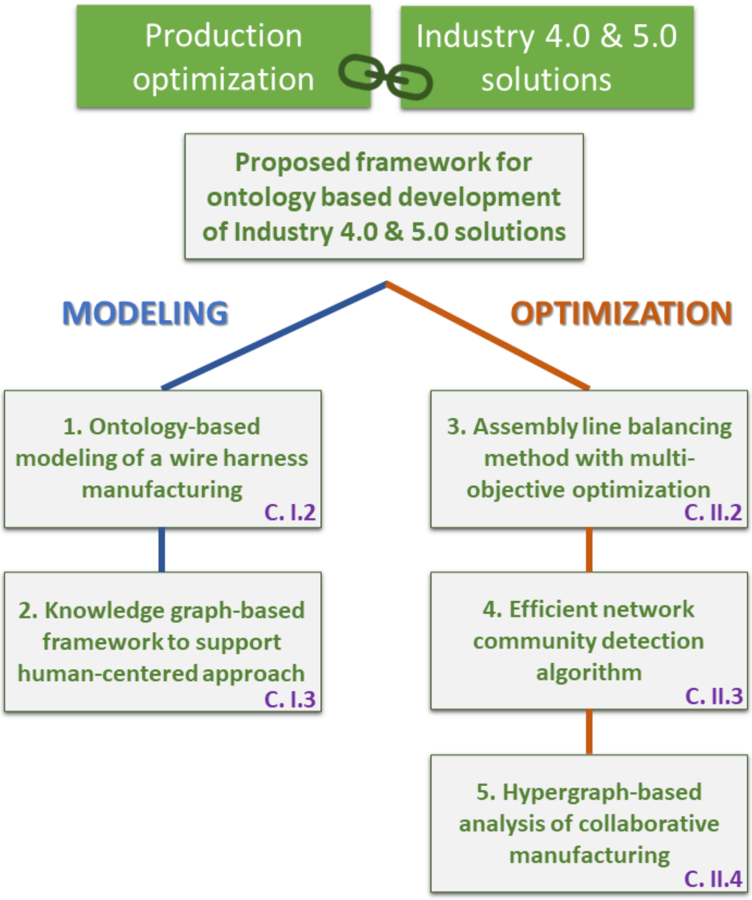


FIGURE B.1: The graphical summary of the thesis findings, divided into modeling and optimization, aims to support the framework for ontology-based development of Industry 4.0 & 5.0 solutions. - The purple titles show the chapter numbers of the related thesis findings.

The following list contains the new scientific results in five thesis findings, numbered in the same way as visualized in Figure B.1:

1. **I highlighted and verified on a wire harness assembly process,** that ontology-based modeling can be utilized to create structured and contextualized models that can support the development of the manufacturing process.

    1.1. I developed an ontology-based framework for analysing industrial processes, using RDF graph data queries. The proposed framework efficiently integrates information based on industrial standards and process data, which is beneficial for data mapping and system analysis.

    1.2. I demonstrated the applicability of ontology-based process modeling and data analysis on a wire-harness assembly-based benchmark, where semantic modeling and data query analysis was performed.

    Related publication: László Nagy, Tamás Ruppert, János Abonyi: *Ontology-Based Analysis of Manufacturing Processes: Lessons Learned from the Case Study of Wire Harness Production*, 2021, Hindawi, Complexity [363]

2. **I have demonstrated that knowledge graph-based framework can support the development of human-centered collaborative and ergonomic manufacturing in Industry 5.0. solutions.**

    2.1. I developed the Human-Centric Knowledge Graph (HCKG) framework, which adapts ontologies and standards and can model operator-related factors, such as monitoring movements, work conditions, or collaboration with robots. The work performed by the operator is in the scope, including the evaluation of movements, collaboration with machines, ergonomics and other conditions.

    2.2. I demonstrated the utilization of the framework in a complex assembly line-based use case, with application examples of resource allocations and comprehensive support of the shop floor workers in collaboration and ergonomics aspects. I verified that knowledge graphs and ontology-based models can support human-centric manufacturing, with several graph-based data queries, visualization, and query analytics presented.

Related conference paper: László Nagy, Tamás Ruppert, János Abonyi: *Human-centered knowledge graph-based design concept for collaborative manufacturing*, 2022 IEEE 27^{th} International Conference on Emerging Technologies and Factory Automation (ETFA) [364]

3. **I introduced a novel, combined analytic hierarchy process and multilayer network-based method for assembly line balancing.**

   3.1. I proposed a multilayer network-based representation of the assembly line-balancing problem, where the layers of the network represent the skills of the operators, the tools required for their activities, and the precedence constraints of their activities. The activity-operator network layer has been designed by a multi-objective optimization algorithm, where the training and equipment costs as well as the precedence of the activities are taken into account.

   3.2. I utilized the analytic hierarchy process technique to evaluate the costs and quantify the importance of the criteria. I verified that the optimization problem can be solved by a multi-level simulated annealing algorithm, that efficiently handles the precedence constraints.

   3.3. I demonstrated the efficiency of the method in a case study of wire harness manufacturing. The applicability of the result relays on more complex assembly lines, where several factors are required to be optimized, and to find the optimal assignment of tasks to operators, aiming to improve the efficiency of the production systems.

Related publication: László Nagy, Tamás Ruppert, János Abonyi: *Analytic Hierarchy Process and Multilayer Network-Based Method for Assembly Line Balancing*, 2020, Applied Sciences [365]

4. **I developed an efficient network community detection algorithm based on crossing minimization and bottom-up segmentation that can facilitate graph-based analytics.**

   4.1. I integrated the barycentric serialization-based co-clustering and bottom-up segmentation methods to get an effective algorithm to detect communities in graph networks. Hence the nodes are efficiently pre-ordered according to their neighbors by barycentric serialization, the segmentation algorithm provides modules in a computationally more efficient way, than the most frequently used Louvain community detection algorithms.

4.2. I verified the developed method by comparing it with other community detection algorithms using benchmark problems. The results have shown that community detection and clustering can be more efficient in the case of large, graph-based datasets with the combined method that can support the ontology-based development of smart factory applications.

5. **I validated a hypergraph-based method for analysis of collaborative manufacturing.**

5.1. I developed a hypergraph-based analysis method to design Industry 5.0 solutions, which requires a problem-specific description of manufacturing systems, the skills, and states of the operators, as well as of the sensors placed in the intelligent space for the simultaneous monitoring of the collaborative work. I highlighted that the proposed method can detect collaboration types within human and robot actors and perform a systematic analysis of the critical sets of interacting elements in collaborative manufacturing.

5.2. I demonstrated on several application examples, that studying the centrality and modularity of the resultant hypergraphs can support the formation of collaboration and interaction schemes and the formation of manufacturing cells. The results proved that the hypergraph-based method can be used for the investigation of collaborative manufacturing and support the analysis of the robustness.

Related publication: László Nagy, Tamás Ruppert, Andreas Löcklin, János Abonyi: *Hypergraph-based analysis and design of intelligent collaborative manufacturing space*, 2022, Journal of Manufacturing Systems [366]

Chapter C

Appendix

This Chapter contains all appendix related to the presented doctoral thesis. First, Section C.1 presents the general version of the wire harness assembly-based industrial case study, followed by the collaborative scenario in Section C.2. Section C.3 presents the detailed structural diagram of the case study-specific knowledge graph of Chapter I.3. In Section C.4, the abbreviations of the developed assembly line balancing algorithm of Chapter II.2 can be found. Finally, in Section C.5, the list of the used nominations and benchmark results, related to the community detection algorithm of Chapter II.3 can be found.

C.1 Wire harness assembly based industrial case study - general

This doctoral thesis has applied an open-source benchmark problem of a modular wire-harness production system; therefore, this section describes the wire harness assembly-based case study, which is studied in Chapters I.2 and II.2. Wire harnesses are produced by a typical complex modular production system [367]. In wire harness manufacturing the operators work with several tools that perform different activities at workstations to manufacture complex cables. In many cases the assembly procedure is designed to be performed not only at fix work stations but on different assembly tables (where several assembly zones are defined), placed to a conveyor system, which is illustrated at Figure C.1.

Figure C.1: A conveyor system of a wire harness assembly line, where parts, connectors, clips, and wires are placed on tables by operators [285]

The case study assumes that it is possible to improve the manufacturing efficiency if the resources, activities, skills and precedence are better designed. The challenge is that the numerous activities and the highly manual assembly necessary require optimum assembly line balancing. The information that needs to be acquired is a precise prediction of the duration of these activities [285], which can be measured by a fixture sensor.

The manufacturing is modular, that means, the products $p_1, \ldots, p_{N_p}$ are built from a set of modules $m_0, \ldots, m_{N_m}$. The number of types of products N_p was 64 and N_m was defined as a combination of 7 modules: m_0 base module, m_1 as left- or right-hand drive, m_2 normal/hybrid, m_3 halogen/LED lights, m_4 petrol/diesel engine, m_5 4 doors/5 doors and m_6 manual or automatic gearbox. N_a was defined as 654 activities/tasks categorized into N_t which consisted of 16 activity types with well-modeled activity times. These times are based on benchmarks from the literature [368].

The types of activities and the related activity times according to wire harness assembly practice [367] are summarized in Table C.1. Furthermore, Table C.1 defines which activity time depends on the number of wires. The activity times are calculated using a direct proportionality approach, e.g. when an operator is laying four wires over one foot, proportionally to the parameter t_4, the activity time will be $1 \times 6.9s + 4 \times 4.2s = 23.7s$.

Table C.1: Details of the activities during the wire harness assembly

ID	**Activity**	**Remark**	**Unit**	**Time [s]**
t_1	Point-to-point wiring on chassis	Direct wiring	Per wire	1.5
t_2	Laying in U-channel			2.0
t_3	Laying flat cable			4.0
t_4	Laying wire(s) onto harness jig	Laying flat cable	Base time	2.5
			Per wire	5.0
t_5	Laying cable connector (one end) onto harness jig	To the same breakout	Base time	3.2
			Per wire	2.3
t_6	Spot-tying onto cable and cutting it with a pair of scissors			3.3
t_7	Lacing activity			1.25
t_8	Taping activity			1.0
t_9	Inserting into tube or sleeve			1.5
t_{10}	Attachment of wire terminal	Terminal-block fastening (fork lug)		6.5
t_{11}	Screw fastening of terminal			7.55
t_{12}	Screw-and-nut fastening of terminal			12.35
t_{13}	Circular connector	Installation only		5.65
t_{14}	Rectangular connector	Latch or snap-on		11.0
t_{15}	Clip installation			4.0

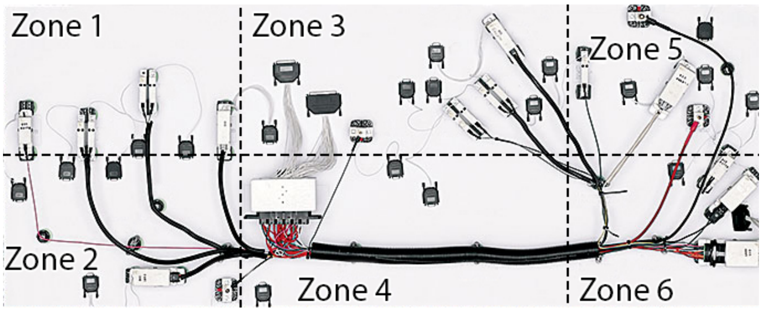


FIGURE C.2: Illustration of the distribution of the fixtures on an assembly table and the definitions of the zones. As the fixtures move according to the assembly tables of the conveyor system, the fixtures are identically placed at every workstation.

In these activities, N_c was equal to 653 different built-in parts (among these $C_r = 299$ terminals, $C_b = 113$ bandages, $C_c = 38$ connectors, $C_d = 155$ wires and $C_l = 48$ clips). N_z was also defined as 6 zones for the workstations (see Figure C.2) to determine where the components are placed on the assembly table.

The assembly line N_w consisted of 10 workstations (assembly tables). For every assembly table, one operator is assigned therefore, $N_o = 10$. The required N_s was also defined as 5 skills of the operators, namely: s_1 - laying cable, s_2 - bandaging, s_3 - attaching the terminal, s_4 - installing the connector and s_5 - inserting the clip. A piece of equipment is required for every activity, therefore $N_e = 5$: e_1 - cabling tool, e_2 - bandaging tool, e_3 - terminal handler, e_4 - connector handler and e_5 - clipping tool. Some tools require a resource ($N_r = 2$): r_1 - compressed air and r_2 - electricity.

Additionally, a subset of this model is described which consists of 24 activities, five operators, six skills and eight pieces of equipment. The elementary activity times that influence the assembly line balance were determined based on expert knowledge [167] (see Table C.2). A more detailed description of the activities, pieces of equipment and skills can be found in Tables C.2 - C.6.

Table C.2: List of the elementary activities with time

Activity ID	**Description**	**Time**
A1	Connector handling	4 s
A2	Connector handling	3 s
A3	Connector handling	2 s
A4	Connector handling	3 s
A5	Insert 1^{st} end + routing	10 s
A6	Insert 2nd end	5 s
A7	Insert 1^{st} end + routing	10 s
A8	Insert 2^{nd} end	5 s
A9	Insert 1^{st} end + routing	10 s
A10	Insert 2^{nd} end	5 s
A11	Insert 1^{st} end + routing	10 s
A12	Insert 2^{nd} end	5 s
A13	Insert 1^{st} end + routing	10 s
A14	Insert 2^{nd} end	5 s
A15	Insert 1^{st} end + routing	10 s
A16	Insert 2^{nd} end	5 s
A17	Insert 1^{st} end + routing	10 s
A18	Insert 2^{nd} end	5 s
A19	Taping	15 s
A20	Taping	13 s
A21	Taping	11 s
A22	Taping	17 s
A23	Taping	15 s
A24	Quality check	10 s

The following tables give a more detailed description of the activities, equipment (Table C.3) and skills (Table C.4) which are involved in the proposed case study. Furthermore, the activity–equipment (Table C.5) and activity–skill (Table C.6) connectivity matrices show the requirements of the given base activity.

Table C.3: List of equipment that can be allocated in the assembly process

Equipment ID	**Description**
E1	Connector fixture
E2	Connector fixture
E3	Routing tool
E4	Insertion tool
E5	Taping tool (expert)
E6	Taping tool (normal)
E7	Taping tool (normal)
E8	Repair tool

Table C.4: Description of skills that can be used in the studied production process

Skill ID	**Description**
S1	Connector handling skill
S2	Insertion (normal) and routing skills
S3	Insertion (expert) skill
S4	Taping (normal) skill
S5	Taping (expert) skill
S6	Quality (expert) skill

Table C.5: Activity–equipment matrix that defines which equipment are required to perform a given activity

	E1	**E2**	**E3**	**E4**	**E5**	**E6**	**E7**	**E8**
A1	1	1						
A2	1	1						
A3	1	1						
A4	1	1						
A5			1					
A6				1				
A7			1					
A8				1				
A9			1					
A10				1				
A11			1					
A12				1				
A13			1					
A14				1				
A15			1					
A16				1				
A17			1					
A18				1				
A19					1			
A20					1			
A21						1	1	
A22						1	1	
A23						1	1	
A24								1

Table C.6: Activity–skill matrix that defines which skills are required to perform a given activity

	S1	**S2**	**S3**	**S4**	**S5**	**S6**
A1	1					
A2	1					
A3	1					
A4	1					
A5		1				
A6			1			
A7		1				
A8			1			
A9		1				
A10			1			
A11		1				
A12			1			
A13		1				
A14			1			
A15		1				
A16			1			
A17		1				
A18			1			
A19				1		
A20				1		
A21					1	
A22					1	
A23					1	
A24						1

The tables illustrate that a complex assembly procedure is also influenced by how much equipment is needed for the designed production line and how many skills should be learned by the operators.

C.2 Wire harness assembly based industrial case study - collaboration

This section describes the wire harness assembly-based case study with collaboration aspects, which is studied in Chapters I.3 and II.4. First, Figure C.3 represents the shop floor grid layout, where a coordinate system provides the optional grid to allocate operators or production resources, such as robots and machines. The case study includes a real-time location system (RTLS) that tracks the position of assembly workers and assets. The X and Y axes correspond to the possible RTLS-based positions on the shop floor. The grid can also provide information about the distances needed for material handling and transportation. Additionally, 18 different areas, e.g. ST_11, are defined, which are capable of providing space for a workstation on the shop floor.

A double production line consisting of batch and conventional production was defined and the process flow is shown in Figure C.4, which is based on a real assembly line from the wire harness manufacturing industry. The process consists of two assembly lines that share tasks and resources. The elements of these production lines are listed in Table C.8. As for the shop floor, two Storage, several Buffers, Crimping stations and Assembly stations are defined. The second group of elements consists of the human-machine members, which can be Operators or Robots, as well as the assets of the production line, namely Machines, Tools,

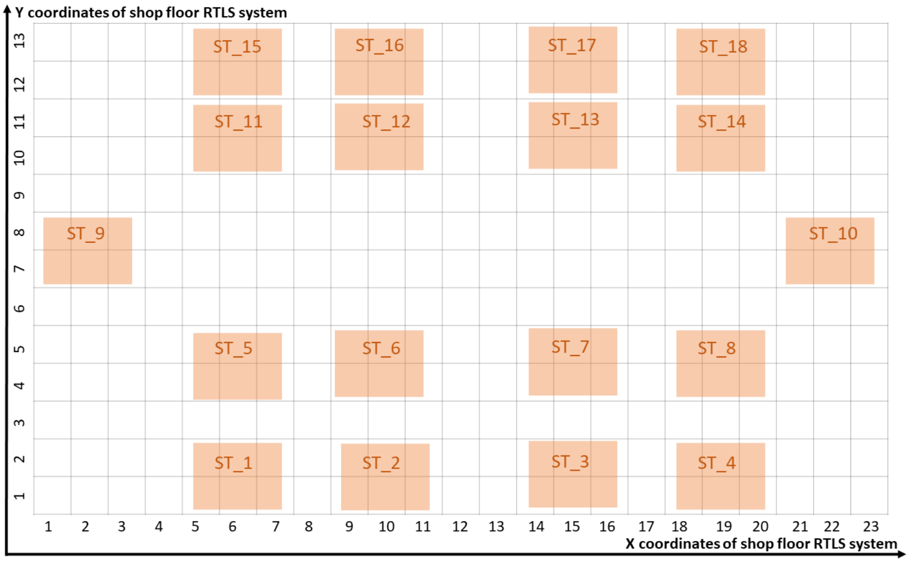


FIGURE C.3: The grid layout of the benchmark shop floor

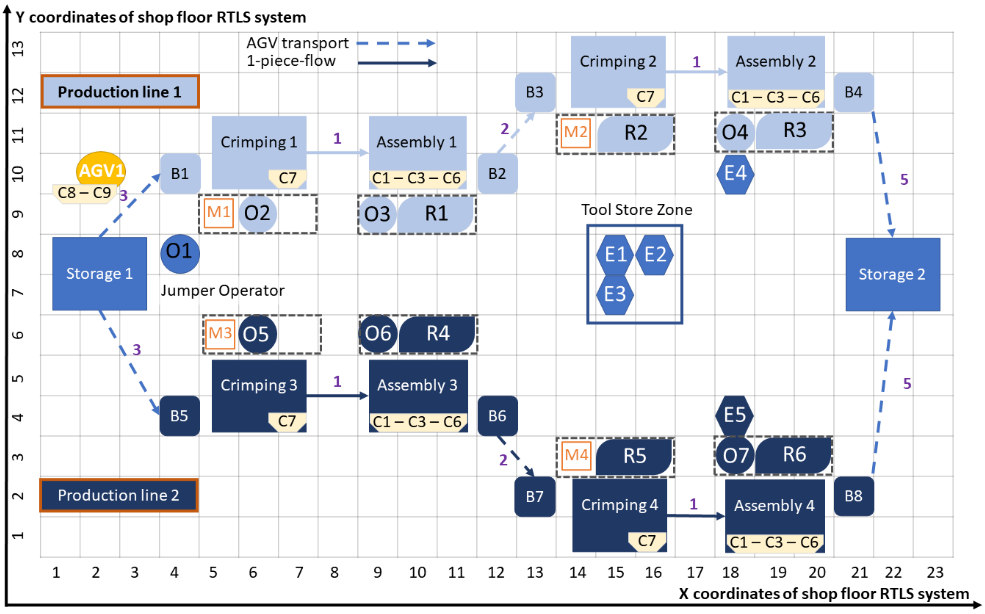


FIGURE C.4: The process flow of the wire harness assembly line benchmark

Screwdrivers and the AGV (Automated Guided Vehicle). Finally, Capabilities are required to perform particular activities and Sensor elements to monitor the collaborative space.

A specific list of activity types for this benchmark problem is given in Table C.7, with categories, e.g. the crimping process, assembly process or material handling, as well as the definitions of the results of these activity types. Defining not only the activity types but their results afterwards is important in terms of tracking the processes and collaboration.

In Figure C.4, the elements denoted in a brighter color represent *Production line 1* and the darker ones *Production line 2*, while in the middle, the shared assets and resources are visualized. The material handling steps during the production process are highlighted with arrows, which can be a one-piece-flow performed by operators or an AGV-based transport system. Additionally, the distances over which material is handled are denoted by purple numbers. The process flow (visualized in Figure C.4) starts at *Storage 1*, where the so-called jumper operator $O1$ loads $AGV1$ (using capability $C8$) with one batch and $AGV1$ transfers it to *Crimping station 1* or *3* (using capability $C9$), where operator $O2$ or $O5$ unloads it into the local buffer $B1$ or $B5$. The following steps are the same on both production lines, moreover, the process description will be continued with *Production line 1*. Based on the production plan, operator $O2$ performs the crimping-related

Table C.7: The activity types in the wire harness assembly process and their results

	Crimping process
t18	Manual handling of a wire from a buffer *Result: One piece of wire is moved to the crimping station from the buffer.*
t19	Positioning of a crimp into a vise *Result: Crimp is positioned into a vise.*
t20	Inserting a wire into a crimp *Result: Wire is inserted into a crimp.*
t21	Starting a machine *Result: Machine is running.*
t22	Crimping *Result: Crimping is finished.*
t23	Manual handling of a semi-finished product *Result: Semi-finished product is removed from the vise.*
t24	Handover of a semi-finished product *Result: Semi-finished product is moved to another station.*
	Assembly process
t2	Laying in a U-channel *Result: U-channel is laid in the right assembly zone.*
t4	Laying wire(s) onto a harness jig *Result: Wire(s) is (are) laid correctly onto a harness jig.*
t9	Insertion into a tube or sleeve *Result: Tube is inserted into the correct sleeve.*
t11	Fastening of the terminal with screws *Result: Terminal screws are fastened.*
t25	Positioning of a crimp into a fixture *Result: Crimp is correctly positioned into the fixture.*
t26	Manual handling of a semi-finished product into a buffer *Result: Semi-finished product is placed into the buffer.*
	Material handling
t16	Loading of the AGV *Result: Parts are loaded on to the rack of the AGV.*
t17	Transportation by an AGV *Result: AGV moved the position from the source to its destination*
t27	Unloading of the AGV *Result: Parts are unloaded from the rack of the AGV.*

activities listed in Table C.7 that require capability $C7$. Furthermore, machine $M1$ is also utilized during these crimping activities. Finally, operator $O2$ hands over the workpiece to operator $O3$ at *Assembly station 1* (one-piece-flow). Operator $O3$ and robot $R1$ collaborate with each other, while capabilities $C1, C3$ and $C6$-related activities are performed. Moreover, tools $E1$-3 are also used during the activity steps of *Assembly station 1.* At the end of the procedure, operator $O3$ places the workpiece into buffer $B2$. If a whole batch has been completed, the same operator loads $AGV1$, which delivers the batch of cables to the next buffer, that is, $B3$. Afterwards robot $R2$ unloads the buffer and performs capability $C7$ and machine $M2$-related activities at *Crimping station 2.* Then robot $R2$ hands over the workpiece (one-piece-flow) to operator $O4$ at the next station, namely *Assembly station 2.* At the last workstation of Production line 1, operator $O4$ and robot $R3$ collaborate with each other to perform activities that require capabilities $C1, C3$ and $C6$. At the end of the assembly line, operator $O4$ places the workpieces into buffer $B4$. If a whole batch has been completed, the same operator loads $AGV1$, which delivers the products to their final destination, namely $Storage$ 2.

Further attributes of the elements (besides the list of main elements in Table C.8) are the following *Capabilities*, which are required to perform special activities: $C1$ - Inserting and laying of parts (cabling), $C3$ - Terminal handling, $C6$ - Fastening the terminal with screws, $C7$ - Operation of the crimping machine, $C8$ - Loading or unloading of the AGV and $C9$ - Transportation of the workpieces on the shop floor. Special tools, which are partly shared assets of the procedure are also present, namely $E1$ - wiring tool, $E2$ - tubing tool and $E3$-$E5$, screwdrivers. Furthermore, several unique *Machines* (M) are allocated to different *Crimping stations* and *Tools* (E) are regarded as shared assets within *Assembly stations.*

Additionally, it is important to mention that different types of sensors are also parts of this case study, whose goal is to make observations about each activity, human and machine member of the production line as well as monitor the working conditions. These groups of sensors are as follows: Camera system, Real-time locating system, Robot-embedded sensor data, Machine-embedded sensor data, Environment sensor shield and Human body sensor.

In the following tables, a more detailed overview of the wire harness assembly benchmark is provided, where first in Table C.9 each activity type of the complex

Table C.8: The elements of the wire harness assembly lines

Work sections of the production lines	
Storage	$[K1, K2]$
Buffer	$[B1, B2, B3, B4, B5, B6, B7, B8]$
Crimping stations	[Crimping 1, Crimping 2, Crimping 3, Crimping 4]
Assembly stations	[Assembly 1, Assembly 2, Assembly 3, Assembly 4]
Human-machine members and assets	
Operators	$[O1, O2, O3, O4, O5, O6, O7]$
Robots	$[R1, R2, R3, R4, R5, R6]$
AGV	$[AGV1]$
Machines	$[M1, M2, M3, M4]$
Tools	$[E1, E2, E3, E4, E5]$
Capabilities	$[C1, C3, C6, C7, C8, C9]$

industrial process is listed, then in Tables C.10-C.11 along with the details of the sequence of activities, which is distinguished in chapters I.3 and II.4.

Table C.9: Description of the different activity types in the entire wire harness assembly benchmark

Activity type ID	**Description of the activity type**
t1	Point-to-point wiring on a chassis
t2	Laying in a U-channel
t3	Laying a flat cable
t4	Laying wire(s) onto the harness jig
t5	Laying one end of a cable connector onto a harness jig
t6	Spot-tying onto a cable and cutting it with a pair of scissors
t7	Lacing activity
t8	Lacing activity
t9	Inserting into a tube or sleeve
t10	Attachment of a wire terminal
t11	Screw fastening of a wire terminal
t12	Screw-and-nut fastening of a wire terminal
t13	Circular connector
t14	Rectangular connector
t15	Clip installation
t16	Loading of the AGV
t17	Transportation
t18	Manual handling of a wire from a buffer
t19	Positioning of a crimp into a vise
t20	Inserting a wire into a crimp
t21	Starting a machine
t22	Crimping
t23	Manual handling of a semi-finished product
t24	Handover of a semi-finished product
t25	Positioning of a crimp into a fixture
t26	Manual handling of a semi-finished product into a buffer
t27	Unloading of the AGV

Table C.10: The sequence of activities as well as the results of the proposed wire harness assembly benchmark and their details - Part 1

Activity ID	**Activity type ID**	**Result ID**	**resultTypeID**	**Process step**	**Number of process step**
a1	t16	res1	res_type_16	Storage 1 - AGV1	1
a2	t17	res2	res_type_17	Storage 1 - Buffer1	1
a3	t27	res3	res_type_27	AGV1 - Buffer1	1
a4	t18	res4	res_type_18	Buffer1 - Crimping1	Batch size
a5	t19	res5	res_type_19	Crimping1	Batch size
a6	t20	res6	res_type_20	Crimping1	Batch size
a7	t21	res7	res_type_21	Crimping1	Batch size
a8	t22	res8	res_type_22	Crimping1	Batch size
a9	t23	res9	res_type_23	Crimping1	Batch size
a10	t24	res10	res_type_24	Crimping1 - Assembly1	Batch size
a11	t24	res10	res_type_24	Crimping1 - Assembly1	Batch size
a12	t25	res11	res_type_25	Assembly1	Batch size
a13	t02	res12	res_type_02	Assembly1	Batch size
a14	t02	res12	res_type_02	Assembly1	Batch size
a15	t04	res13	res_type_04	Assembly1	Batch size
a16	t04	res13	res_type_04	Assembly1	Batch size
a17	t09	res14	res_type_09	Assembly1	Batch size
a18	t09	res14	res_type_09	Assembly1	Batch size
a19	t11	res15	res_type_11	Assembly1	Batch size
a20	t11	res15	res_type_11	Assembly1	Batch size
a21	t26	res16	res_type_26	Assembly1 - Buffer2	Batch size
a22	t16	res17	res_type_16	Buffer2 - AGV1	1
a23	t17	res18	res_type_17	Buffer2 - Buffer3	1
a24	t27	res19	res_type_27	AGV1 - Buffer3	1
a25	t18	res20	res_type_18	Buffer3 - Crimping2	Batch size
a26	t19	res21	res_type_19	Crimping2	Batch size
a27	t20	res22	res_type_20	Crimping2	Batch size
a28	t21	res23	res_type_21	Crimping2	Batch size
a29	t22	res24	res_type_22	Crimping2	Batch size
a30	t23	res25	res_type_23	Crimping2	Batch size
a31	t24	res26	res_type_24	Crimping2 - Assembly2	Batch size
a32	t24	res26	res_type_24	Crimping2 - Assembly2	Batch size
a33	t25	res27	res_type_25	Assembly2	Batch size
a34	t02	res28	res_type_02	Assembly2	Batch size
a35	t02	res28	res_type_02	Assembly2	Batch size
a36	t04	res29	res_type_04	Assembly2	Batch size
a37	t09	res30	res_type_09	Assembly2	Batch size
a38	t11	res31	res_type_11	Assembly2	Batch size
a39	t11	res31	res_type_11	Assembly2	Batch size
a40	t26	res32	res_type_26	Assembly2 - Buffer4	Batch size
a41	t16	res33	res_type_16	Buffer4 - AGV1	1
a42	t17	res34	res_type_17	Buffer4 - Buffer9	1
a43	t27	res35	res_type_27	AGV1 - Storage 2	1
a44	t16	res36	res_type_16	Storage 1 - AGV1	1
a45	t17	res37	res_type_17	Storage 1 - Buffer5	1
a46	t27	res38	res_type_27	AGV1 - Buffer5	1
a47	t18	res39	res_type_18	Buffer5 - Crimping3	Batch size
a48	t19	res40	res_type_19	Crimping3	Batch size
a49	t20	res41	res_type_20	Crimping3	Batch size
a50	t21	res42	res_type_21	Crimping3	Batch size
a51	t22	res43	res_type_22	Crimping3	Batch size
a52	t23	res44	res_type_23	Crimping3	Batch size
a53	t24	res45	res_type_24	Crimping3 - Assembly3	Batch size
a54	t24	res45	res_type_24	Crimping3 - Assembly3	Batch size
a55	t25	res46	res_type_25	Assembly3	Batch size
a56	t02	res47	res_type_02	Assembly3	Batch size
a57	t02	res47	res_type_02	Assembly3	Batch size
a58	t04	res48	res_type_04	Assembly3	Batch size
a59	t04	res48	res_type_04	Assembly3	Batch size
a60	t09	res49	res_type_09	Assembly3	Batch size
a61	t09	res49	res_type_09	Assembly3	Batch size
a62	t11	res50	res_type_11	Assembly3	Batch size
a63	t11	res50	res_type_11	Assembly3	Batch size
a64	t26	res51	res_type_26	Assembly3 - Buffer6	Batch size
a65	t16	res52	res_type_16	Buffer6 - AGV1	1
a66	t17	res53	res_type_17	Buffer6 - Buffer7	1
a67	t27	res54	res_type_27	AGV1 - Buffer7	1
a68	t18	res55	res_type_18	Buffer7 - Crimping4	Batch size
a69	t19	res56	res_type_19	Crimping4	Batch size
a70	t20	res57	res_type_20	Crimping4	Batch size
a71	t21	res58	res_type_21	Crimping4	Batch size
a72	t22	res59	res_type_22	Crimping4	Batch size
a73	t23	res60	res_type_23	Crimping4	Batch size

Table C.11: The sequence of activities as well as the results of the proposed wire harness assembly benchmark and their details - Part 2

Activity ID	**Activity type ID**	**Result ID**	**resultTypeID**	**Process step**	**Number of process step**
a74	t24	res61	res_type_24	Crimping4 - Assembly4	Batch size
a75	t24	res61	res_type_24	Crimping4 - Assembly4	Batch size
a76	t25	res62	res_type_25	Assembly4	Batch size
a77	t02	res63	res_type_02	Assembly4	Batch size
a78	t02	res63	res_type_02	Assembly4	Batch size
a79	t04	res64	res_type_04	Assembly4	Batch size
a80	t09	res65	res_type_09	Assembly4	Batch size
a81	t11	res66	res_type_11	Assembly4	Batch size
a82	t11	res66	res_type_11	Assembly4	Batch size
a83	t26	res67	res_type_26	Assembly4 - Buffer8	Batch size
a84	t16	res68	res_type_16	Buffer8 - AGV1	1
a85	t17	res69	res_type_17	Buffer8 - Buffer9	1
a86	t27	res70	res_type_27	AGV1 - Storage 2	1

C.3 Detailed structural diagram of the case study specific KG

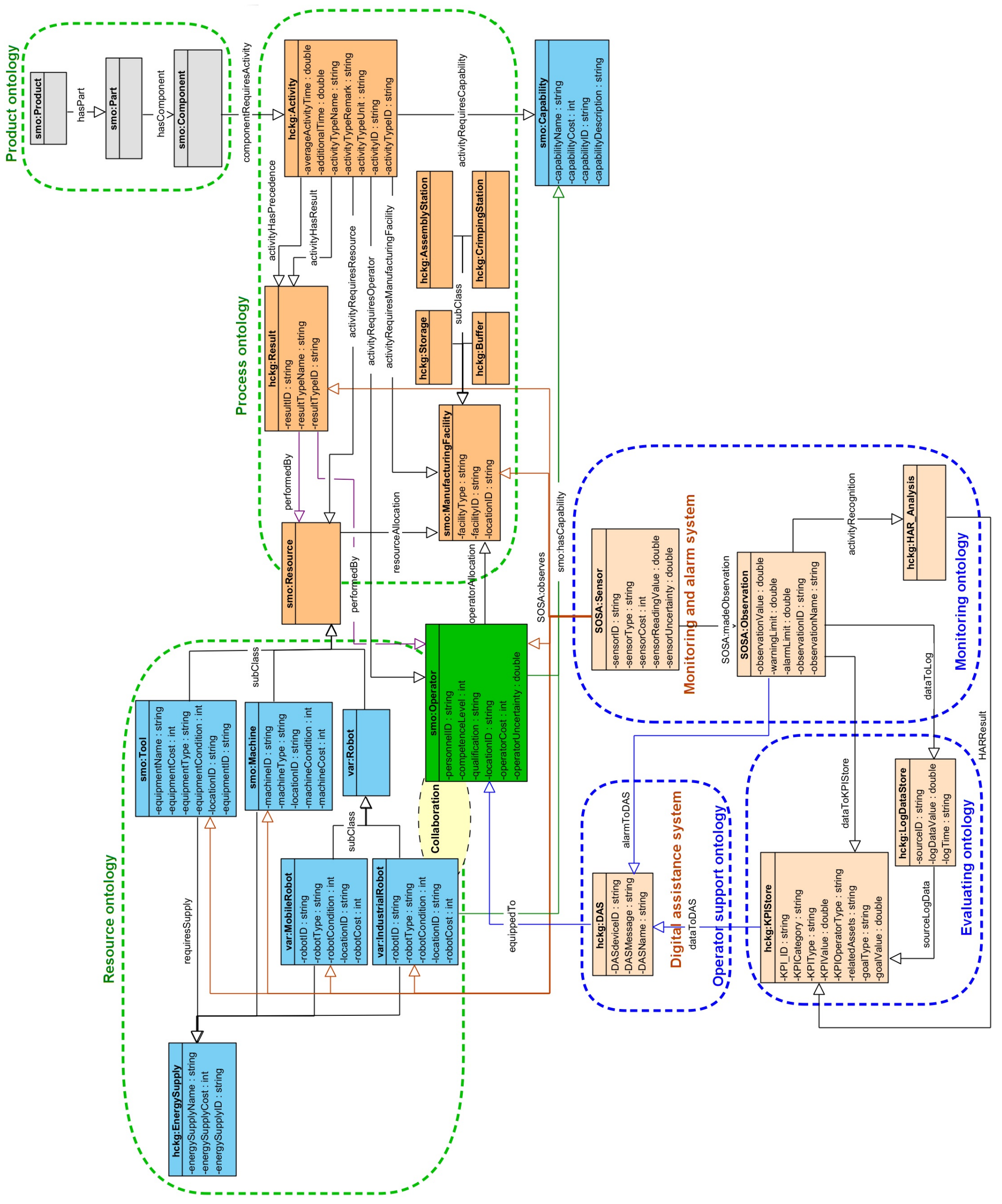


Figure C.5: The fully detailed structural diagram of the developed wire harness assembly specific KG

C.4 Assembly line balancing algorithm - Nominations

The following nominations are used in Chapter II.2:

SA	Simulated annealing
ALB	Assembly line balancing
$G_{i,j}$	Bipartite graphs between the ith and jth sets of objects
O_j, O_j	General representation of a set of objects as $O_i, O_j \in \{\mathbf{s}, \mathbf{e}, \mathbf{a}^{'}, \mathbf{a}, \mathbf{w}\}$
$\mathbf{a} = a_1, \ldots, a_{N_a}$	Index of activities
$\mathbf{o} = o_1, \ldots, o_{N_o}$	Index of operators
$\mathbf{s} = s_1, \ldots, s_{N_s}$	Index of skills
$\mathbf{e} = e_1, \ldots, e_{N_e}$	Index of equipment
$\mathbf{w} = w_1, \ldots, w_{N_w}$	Index of workstations
$\mathbf{W}$	Workstation assigned for the activity, $N_a \times N_w$
$\mathbf{O}$	Operators assigned for the activity, $N_a \times N_o$
$\mathbf{S}$	Skills assigned for the activity, $N_a \times N_s$
$\mathbf{E}$	Equipment assigned for the activity, $N_a \times N_e$
$\mathbf{A}'$	Precedence constraint between activities, $N_a \times N_a$
$\mathbf{T} = t_1, \ldots, t_{N_a}$	Activity time
c_1	Station-time-related cost
c_2	Skill-related (training) cost
c_3	Equipment-related cost
T_c	Cycle time
$\overline{T_c}$	Mean cycle time of N_o operators
N_w	Number of workstations
N_o	Number of operators
N_s	Number of skills
N_e	Number of pieces of equipment
N_π	Number of sequence elements

C.5 Community detection - List of the used nominations and benchmark results

List of the used nominations in Chapter II.3:

$\mathbf{A}$	Adjacency matrix of graph network (real wiring diagram of a network)
$a_{i,j}$, $a_{j,i}$	Node elements of the adjacency matrix (0 or 1)
N	Number of nodes in a network/graph
L	Links or edges of the graph
C	Number of communities
C_c	One of the communities /strongly connected set of nodes/ in a network
N_c	Number of nodes within a community
L_c	Number of links which connect the nodes within a community
M	The total modularity of the network
M_c	The modularity value of a c community
k_c	Total degree of the nodes in the community C_c
$p_{i,j}$	The degree preserving null model
k_i, k_j	Node degrees
$\mathbf{b}$, b_i	The barycentric coordinate vector and the i-th element
$\mathbf{x}$, x_i, x_j	Sorted/serialised order of nodes and the i-th or j-th element
$\mathbf{k}$	Degree of each node in $\mathbf{A}$ adjacency matrix
ω_i	Weight of the i-th node
$\tilde{\mathbf{x}_{old}}$	The $\mathbf{x}$ node order from the previous iteration
α	Stopping criteria adjustment value
$maxIter$	Maximum number of iteration in the crossing minimization algorithm
$\Delta M_{c,c_{c+1}}$	The modularity after merge c and $c+1$ communities
k_c, k_{c+1}	Total degree values within c and $c+1$ communities
L_c, L_{c+1}	The total number of links within communities c and $c+1$
$l_{c,c_{c+1}}$	The number of direct links between the nodes of communities c and $c+1$
$L_{c,c_{c+1}}$	Number of links after merge c and $c+1$ communities
$k_{c,c_{c+1}}$	Total degree value after merge c and $c+1$ communities
γ	Tuning operator to handle the resolution limit problem
l, r	The boundaries of the communities
l_c, r_c	The left and the right segment boundary of the c-th community
$mergecost$	The 'benefit' of merging together the c-th and the $c+1$-th segment

TABLE C.12: Modularity based community detection benchmark

Network name	Node	Res.	γ	Proposed method			GCDanon			GCModulMax3			GCModulMax2			GCModulMax1			GCReichardt		
				Mod.	C	t [sec]	Mod.	C	t [sec]	Mod.	C	t [sec]	Mod.	C	t [sec]	Mod.	C	t [sec]	Mod.	C	t [sec]
karate	34	5	2,5	0,671	3	0,035	0,650	4	0,017	0,662	3	0,014	0,653	4	0,073	0,671	4	0,071	0,662	3	0,012
dolphins	62	5	2,5	0,691	3	0,009	0,713	4	0,005	0,741	4	0,005	0,697	5	0,022	0,702	5	0,024	0,741	4	0,003
lesmis	77	5	2,5	0,654	4	0,009	0,713	6	0,005	0,674	5	0,004	0,713	6	0,021	0,716	7	0,031	0,674	5	0,003
polbooks	105	5	2,5	0,765	3	0,005	0,810	4	0,007	0,811	4	0,005	0,810	4	0,042	0,806	4	0,020	0,814	4	0,009
football	115	5	2,5	0,649	4	0,013	0,688	7	0,013	0,686	6	0,008	0,702	8	0,034	0,698	9	0,023	0,721	6	0,010
celegansneural	297	10	1,7	0,596	4	0,006	0,573	5	0,042	0,618	5	0,036	0,574	5	0,204	0,531	7	0,111	0,618	5	0,056
email-Eu-core	1.005	75	1,5	0,619	3	0,066	0,585	26	1,590	0,601	44	1,482	0,547	27	0,460	0,566	26	0,514	0,602	47	1,459
graph-1964804849	2.400	75	1,5	0,464	4	0,054	0,256	12	27,6	0,387	15	27,3	0,311	14	1,998	0,308	16	4,868	0,389	17	27,2
graph-1337233344	4.800	75	1,5	0,522	5	0,163	0,325	7	249	0,392	6	250	0,419	6	2,949	0,411	7	9,285	0,441	4	250
inf-power	4.941	75	1,5	0,891	6	0,140	0,958	42	273	0,957	41	274	0,959	35	2,941	0,781	41	82,6	0,960	41	274
citDBLP	12.591	75	1,5	0,678	7	0,476	0,726	69	5.128	0,732	100	5.163	0,711	58	30,3	0,700	62	97,0	0,743	104	5.015
caAstroPh	17.903	75	1,5	0,733	3	0,595	0,682	39	15.304	0,713	168	15.413	0,687	30	73,7	0,658	36	331	0,709	173	15.603
socepinions	26.588	75	1,5	0,620	9	0,962	0,671	77	52.983	0,729	797	55.457	0,694	236	194	0,674	270	1.879	0,696	848	54.593

Bibliography

[1] Michael Rüßmann, Markus Lorenz, Philipp Gerbert, Manuela Waldner, Jan Justus, Pascal Engel, and Michael Harnisch. Industry 4.0: The future of productivity and growth in manufacturing industries. Technical Report 9, Boston Consulting Group, 2015.

[2] Arkadiusz Gola. Reliability analysis of reconfigurable manufacturing system structures using computer simulation methods. *Eksploatacja i Niezawodność*, 21(1), 2019.

[3] Gergely Marcell Honti and Janos Abonyi. A review of semantic sensor technologies in internet of things architectures. *Complexity*, 2019.

[4] Dávid Gyulai, Júlia Bergmann, Viola Gallina, and Alexander Gaal. Towards a connected factory: Shop-floor data analytics in cyber-physical environments. *Procedia Cirp*, 86:37–42, 2019.

[5] Szilárd Jaskó, Adrienn Skrop, Tibor Holczinger, Tibor Chován, and János Abonyi. Development of manufacturing execution systems in accordance with industry 4.0 requirements: A review of standard-and ontology-based methodologies and tools. *Computers in Industry*, 123:103300, 2020.

[6] Zhenjun Ming, Gehendra Sharma, Janet K Allen, and Farrokh Mistree. An ontology for representing knowledge of decision interactions in decision-based design. *Computers in Industry*, 114:103145, 2020.

[7] Csaba Kardos, Gergely Popovics, Botond Kádár, and László Monostori. Methodology and data-structure for a uniform system's specification in simulation projects. *Procedia CIRP*, 7:455–460, 2013.

[8] Wil Van Der Aalst. Process mining. *Communications of the ACM*, 55(8):76–83, 2012.

[9] Ismail Yürek, Derya Birant, and KÖKTEN ULAŞ BİRANT. Interactive process miner: a new approach for process mining. *Turkish Journal of Electrical Engineering & Computer Sciences*, 26(3):1314–1328, 2018.

[10] Jakob Bönsch, Matthes Elstermann, Andreas Kimmig, and Jivka Ovtcharova. A subject-oriented reference model for digital twins. *Computers & Industrial Engineering*, page 108556, 2022.

[11] Fei Tao, Jiangfeng Cheng, Qinglin Qi, Meng Zhang, He Zhang, and Fangyuan Sui. Digital twin-driven product design, manufacturing and service with big data. *The International Journal of Advanced Manufacturing Technology*, 94(9):3563–3576, 2018.

[12] Wael M Mohammed, Rodolfo E Haber, and Jose L Martinez Lastra. Ontology-driven guidelines for architecting digital twins in factory automation applications. *Machines*, 10(10):861, 2022.

[13] Jiaru Bai, Liwei Cao, Sebastian Mosbach, Jethro Akroyd, Alexei A Lapkin, and Markus Kraft. From platform to knowledge graph: evolution of laboratory automation. *JACS Au*, 2(2):292–309, 2022.

[14] Christian Fürber. Semantic technologies. In *Data Quality Management with Semantic Technologies*, pages 56–68. Springer, 2016.

[15] María del Mar Roldán-García, José García-Nieto, Alejandro Maté, Juan Trujillo, and José F Aldana-Montes. Ontology-driven approach for kpi meta-modelling, selection and reasoning. *International Journal of Information Management*, 58:102018, 2021.

[16] Enrique Coronado, Takuya Kiyokawa, Gustavo A Garcia Ricardez, Ixchel G Ramirez-Alpizar, Gentiane Venture, and Natsuki Yamanobe. Evaluating quality in human-robot interaction: A systematic search and classification of performance and human-centered factors, measures and metrics towards an industry 5.0. *Journal of Manufacturing Systems*, 63:392–410, 2022.

[17] Jan Portisch, Sven Hertling, and Heiko Paulheim. Visual analysis of ontology matching results with the melt dashboard. In *European Semantic Web Conference*, pages 186–190. Springer, 2020.

[18] Emna Ammar El Hadj Amor and Sonia Ayachi Ghannouchi. Toward an ontology-based model of key performance indicators for business process improvement. In *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*, pages 148–153. IEEE, 2017.

[19] Soujanya Mantravadi, Charles Møller, LI Chen, and Reto Schnyder. Design choices for next-generation iiot-connected mes/mom: An empirical study on smart factories. *Robotics and Computer-Integrated Manufacturing*, 73:102225, 2022.

[20] Xun Xu, Yuqian Lu, Birgit Vogel-Heuser, and Lihui Wang. Industry 4.0 and industry 5.0—inception, conception and perception. *Journal of Manufacturing Systems*, 61:530–535, 2021.

[21] European Commission, Directorate-General for Research, Innovation, M Breque, L De Nul, and A Petridis. *Industry 5.0 : towards a sustainable, human-centric and resilient European industry.* Publications Office, 2021.

[22] Tuan-anh Tran, Tamas Ruppert, Gyorgy Eigner, and Janos Abonyi. Retrofitting-based development of brownfield industry 4.0 and industry 5.0 solutions. *IEEE Access*, 2022.

[23] Sandra Grabowska, Sebastian Saniuk, and Bożena Gajdzik. Industry 5.0: improving humanization and sustainability of industry 4.0. *Scientometrics*, pages 1–28, 2022.

[24] Saeid Nahavandi. Industry 5.0—a human-centric solution. *Sustainability*, 11(16):4371, 2019.

[25] Vitaly V Martynov, Diana N Shavaleeva, and Alena A Zaytseva. Information technology as the basis for transformation into a digital society and industry 5.0. In *2019 International Conference" Quality Management, Transport and Information Security, Information Technologies"(IT&QM&IS)*, pages 539–543. IEEE, 2019.

[26] Min Zhao. Understanding of a new generation of intelligent manufacturing based on rami 4.0. *Strategic Study of Chinese Academy of Engineering*, 20(4):90–96, 2018.

[27] Jörg Krüger, Terje K Lien, and Alexander Verl. Cooperation of human and machines in assembly lines. *CIRP annals*, 58(2):628–646, 2009.

[28] Daisuke Tsutsumi, Dávid Gyulai, Emma Takács, Júlia Bergmann, Youichi Nonaka, and Kikuo Fujita. Personalized work instruction system for revitalizing human-machine interaction. *Procedia CIRP*, 93:1145–1150, 2020.

[29] Jan Reimann and Gabor Sziebig. The intelligent factory space–a concept for observing, learning and communicating in the digitalized factory. *IEEE Access*, 7:70891–70900, 2019.

[30] Jung-Sing Jwo, Ching-Sheng Lin, and Cheng-Hsiung Lee. Smart technology–driven aspects for human-in-the-loop smart manufacturing. *The International Journal of Advanced Manufacturing Technology*, 114(5):1741–1752, 2021.

[31] Baicun Wang, Pai Zheng, Yue Yin, Albert Shih, and Lihui Wang. Toward human-centric smart manufacturing: A human-cyber-physical systems (hcps) perspective. *Journal of Manufacturing Systems*, 63:471–490, 2022.

[32] Yan M Yufik, Thomas B Sheridan, and Valery F Venda. Knowledge measurement, cognitive complexity, and cybernetics of mutual man–machine adaptation. *Cybernetics and applied systems*, pages 187–238, 2018.

[33] Yuqian Lu, Hao Zheng, Saahil Chand, Wanqing Xia, Zengkun Liu, Xun Xu, Lihui Wang, Zhaojun Qin, and Jinsong Bao. Outlook on human-centric manufacturing towards industry 5.0. *Journal of Manufacturing Systems*, 62:612–627, 2022.

[34] Aditya Akundi, Daniel Euresti, Sergio Luna, Wilma Ankobiah, Amit Lopes, and Immanuel Edinbarough. State of industry 5.0—analysis and identification of current research trends. *Applied System Innovation*, 5(1):27, 2022.

[35] Praveen Kumar Reddy Maddikunta, Quoc-Viet Pham, B Prabadevi, Natarajan Deepa, Kapal Dev, Thippa Reddy Gadekallu, Rukhsana Ruby, and Madhusanka Liyanage. Industry 5.0: A survey on enabling technologies and potential applications. *Journal of Industrial Information Integration*, 26:100257, 2022.

[36] Usharani Hareesh Govindarajan, Amy JC Trappey, and Charles V Trappey. Immersive technology for human-centric cyberphysical systems in complex

manufacturing processes: a comprehensive overview of the global patent profile using collective intelligence. *Complexity*, 2018, 2018.

[37] David Romero, Johan Stahre, Thorsten Wuest, Ovidiu Noran, Peter Bernus, Åsa Fast-Berglund, and Dominic Gorecky. Towards an operator 4.0 typology: a human-centric perspective on the fourth industrial revolution technologies. In *proceedings of the international conference on computers and industrial engineering (CIE46), Tianjin, China*, pages 29–31, 2016.

[38] D Romero and J Stahre. Towards the resilient operator 5.0: The future of work in smart resilient manufacturing systems. In *54th CIRP Conference on Manufacturing Systems*, pages 1089–1094, 2021.

[39] Dimitris Mourtzis, John Angelopoulos, and Nikos Panopoulos. Operator 5.0: A survey on enabling technologies and a framework for digital manufacturing based on extended reality. *Journal of Machine Engineering*, 22, 2022.

[40] Agnieszka Radziwon, Arne Bilberg, Marcel Bogers, and Erik Skov Madsen. The smart factory: exploring adaptive and flexible manufacturing solutions. *Procedia engineering*, 69:1184–1190, 2014.

[41] Elvis Hozdić. Smart factory for industry 4.0: A review. *International Journal of Modern Manufacturing Technologies*, 7(1):28–35, 2015.

[42] Larry González and Aidan Hogan. Modelling dynamics in semantic web knowledge graphs with formal concept analysis. In *Proceedings of the 2018 World Wide Web Conference*, pages 1175–1184, 2018.

[43] Alexander Maedche and Steffen Staab. Ontology learning for the semantic web. *IEEE Intelligent systems*, 16(2):72–79, 2001.

[44] Renzo Angles, Marcelo Arenas, Pablo Barceló, Aidan Hogan, Juan Reutter, and Domagoj Vrgoč. Foundations of modern query languages for graph databases. *ACM Computing Surveys (CSUR)*, 50(5):1–40, 2017.

[45] Nada Sahlab, Simon Kamm, Timo Müller, Nasser Jazdi, and Michael Weyrich. Knowledge graphs as enhancers of intelligent digital twins. In *2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS)*, pages 19–24. IEEE, 2021.

[46] Gunnar Stefansson. Business-to-business data sharing: A source for integration of supply chains. *International journal of production economics*, 75(1-2):135–146, 2002.

[47] Oluwatosin Ahmed Amodu and Mohamed Othman. Machine-to-machine communication: An overview of opportunities. *Computer Networks*, 145:255–276, 2018.

[48] Georg Ferdinand Schneider, Hendro Wicaksono, and Jivka Ovtcharova. Virtual engineering of cyber-physical automation systems: The case of control logic. *Advanced Engineering Informatics*, 39:127–143, 2019.

[49] Lan Yang, Kathryn Cormican, and Ming Yu. Ontology-based systems engineering: A state-of-the-art review. *Computers in Industry*, 111:148–171, 2019.

[50] Thorsten Liebig, Andreas Maisenbacher, Michael Opitz, Jan R Seyler, Gunther Sudra, and Jens Wissmann. Building a knowledge graph for products and solutions in the automation industry. *Computer Science*, 2019.

[51] Jay Pujara, Hui Miao, Lise Getoor, and William Cohen. Knowledge graph identification. In *International Semantic Web Conference*, pages 542–557. Springer, 2013.

[52] Evan Wallace, Dimitris Kiritsis, Barry Smith, Chris Will, et al. The industrial ontologies foundry proof-of-concept project. In *IFIP International Conference on Advances in Production Management Systems*, pages 402–409. Springer, 2018.

[53] Mohamed Karray, Neil Otte, Rahul Rai, Farhad Ameri, Boonserm Kulvatunyou, Barry Smith, Dimitris Kiritsis, Chris Will, Rebecca Arista, et al. The industrial ontologies foundry (iof) perspectives. 2021.

[54] Stefano Borgo, Francesco Compagno, Nicola Guarino, Claudio Masolo, and Emilio M Sanfilippo. An overview of some ontological challenges inengineering maintenance. In *Domain Ontologies for Research Data Management in Industry Commons of Materials and Manufacturing*, 2021.

[55] Csaba Kardos, András Kovács, and József Váncza. A constraint model for assembly planning. *Journal of Manufacturing Systems*, 54:196–203, 2020.

[56] Manuel Fechter, Carsten Seeber, and Shengjian Chen. Integrated process planning and resource allocation for collaborative robot workplace design. *Procedia CIRP*, 72:39–44, 2018.

[57] Lisa Ehrlinger and Wolfram Wöß. Towards a definition of knowledge graphs. *SEMANTiCS (Posters, Demos, SuCCESS)*, 48(1-4):2, 2016.

[58] Gartner and Kasey Panetta. 5 trends drive the gartner hype cycle for emerging technologies.

[59] Gartner and Kasey Panetta. Gartner top 10 strategic technology trends for 2020, 2019. `https://www.gartner.com/smarterwithgartner/gartner-top-10-strategic-technology-trends-for-2020/`.

[60] Martin Hankel and Bosch Rexroth. The reference architectural model industrie 4.0 (rami 4.0). *ZVEI, April*, 410, 2015.

[61] Paulo Leitao, Armando W Colombo, and Francisco J Restivo. Adacor: A collaborative production automation and control architecture. *IEEE Intelligent Systems*, 20(1):58–66, 2005.

[62] M Hannah, C Leiva, and D Noller. The importance of standards in smart manufacturing. *MESA International White Paper*, 58, 2018.

[63] Hakki Ozgur Unver. An isa-95-based manufacturing intelligence system in support of lean initiatives. *The International Journal of Advanced Manufacturing Technology*, 65(5-8):853–866, 2013.

[64] Dazhuang He, Andrei Lobov, and JL Martinez Lastra. Isa-95 tool for enterprise modeling. *Proceeding of ICONS*, pages 83–87, 2012.

[65] Hervé Panetto, Salah Baïna, and Gérard Morel. Mapping the iec 62264 models onto the zachman framework for analysing products information traceability: a case study. *Journal of Intelligent Manufacturing*, 18(6):679–698, 2007.

[66] Grady Booch. *The unified modeling language user guide*. Pearson Education India, 2005.

[67] ANSI/ISA-95.00.03. Enterprise - control system integration. Part 3: activity models of manufacturing operations management. Standard, International Society of Automation, 2005.

[68] Shi-Wan Lin, Brett Murphy, Erich Clauer, Ulrich Loewen, Ralf Neubert, Gerd Bachmann, Madhusudan Pai, and Martin Hankel. Architecture alignment and interoperability: An industrial internet consortium and plattform industrie 4.0 joint whitepaper. *White Paper, Industrial Internet Consortium*, 2017.

[69] Hansong Xu, Wei Yu, David Griffith, and Nada Golmie. A survey on industrial internet of things: A cyber-physical systems perspective. *IEEE Access*, 6:78238–78259, 2018.

[70] Karsten Schweichhart. Reference architectural model industrie 4.0 (rami 4.0). 40, 2016.

[71] Industrial automation systems and integration — integration of life-cycle data for process plants including oil and gas production facilities — part 13: Integrated asset planning life-cycle. Standard, Organization (ISO), 2018.

[72] Rafael Batres, Matthew West, David Leal, David Price, Katsube Masaki, Yukiyasu Shimada, Tetsuo Fuchino, and Yuji Naka. An upper ontology based on iso 15926. *Computers & Chemical Engineering*, 31(5-6):519–534, 2007.

[73] Iec 62264-1:2013 enterprise-control system integration — part 1: Models and terminology. Standard, Organization (ISO), 2013.

[74] Automation systems and integration — digital twin framework for manufacturing — part 1: Overview and general principles. Standard, Organization (ISO), 2021.

[75] Guodong Shao et al. Use case scenarios for digital twin implementation based on iso 23247. *National Institute of Standards: Gaithersburg, MD, USA*, 2021.

[76] Valeriy Vyatkin. The iec 61499 standard and its semantics. *IEEE Industrial Electronics Magazine*, 3(4):40–48, 2009.

[77] Wang Meilin, Zhang Xiangwei, and Dai Qingyun. An integration methodology based on soa to enable real-time closed-loop mrp between mes and erp. In *2010 International Conference on Computing, Control and Industrial Engineering*, volume 1, pages 101–105. IEEE, 2010.

[78] Izaskun Fernandez, Patricia Casla, Iker Esnaola, Laure Parigot, and Angelo Marguglio. Towards adaptive, interactive, assistive and collaborative assembly workplaces through semantic technologies. *Preprint*, 2020.

[79] Rainer Drath, Arndt Luder, Jorn Peschke, and Lorenz Hundt. Automationml-the glue for seamless automation engineering. In *2008 IEEE International Conference on Emerging Technologies and Factory Automation*, pages 616–623. IEEE, 2008.

[80] Bernhard Wally, Christian Huemer, Alexandra Mazak, and Manuel Wimmer. Iec 62264-2 for automationml. In *Proc. 5th AutomationML User Conference*, pages 1–7, 2018.

[81] Manuel Fechter and Alexander Neb. From 3d product data to hybrid assembly workplace generation using the automationml exchange file format. *Procedia CIRP*, 81:57–62, 2019.

[82] Miriam Schleipen and Rainer Drath. Three-view-concept for modeling process or manufacturing plants with automationml. In *2009 IEEE Conference on Emerging Technologies & Factory Automation*, pages 1–4. IEEE, 2009.

[83] Ian Horrocks, Bijan Parsia, Peter Patel-Schneider, and James Hendler. Semantic web architecture: Stack or two towers? In *International Workshop on Principles and Practice of Semantic Web Reasoning*, pages 37–41. Springer, 2005.

[84] Michael Kifer, Jos De Bruijn, Harold Boley, and Dieter Fensel. A realistic architecture for the semantic web. In *International Workshop on Rules and Rule Markup Languages for the Semantic Web*, pages 17–29. Springer, 2005.

[85] Ioan Szilagyi and Patrice Wira. Ontologies and semantic web for the internet of things-a survey. In *IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society*, pages 6949–6954. IEEE, 2016.

[86] Dan Brickley, Ramanathan V Guha, and Andrew Layman. Resource description framework (rdf) schema specification. *W3C*, 1999.

[87] Deborah L McGuinness, Frank Van Harmelen, et al. Owl web ontology language overview. *W3C recommendation*, 10(10):2004, 2004.

[88] Andreas Brodt, Daniela Nicklas, and Bernhard Mitschang. Deep integration of spatial query processing into native rdf triple stores. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 33–42, 2010.

[89] Pouya Zangeneh and Brenda McCabe. Ontology-based knowledge representation for industrial megaprojects analytics using linked data and the semantic web. *Advanced Engineering Informatics*, 46:101164, 2020.

[90] Pascal Hitzler, Markus Krötzsch, Bijan Parsia, Peter F Patel-Schneider, Sebastian Rudolph, et al. Owl 2 web ontology language primer. *W3C recommendation*, 27(1):123, 2009.

[91] Jorge Pérez, Marcelo Arenas, and Claudio Gutierrez. Semantics and complexity of sparql. *ACM Transactions on Database Systems (TODS)*, 34(3):1–45, 2009.

[92] Oussama Meski, Farouk Belkadi, Florent Laroche, and Benoit Furet. Towards a knowledge-based framework for digital chain monitoring within the industry 4.0 paradigm. *Procedia CIRP*, 84:118–123, 2019.

[93] Sebastian R Bader, Irlán Grangel-González, Mayesha Tasnim, and Steffen Lohmann. Structuring the industry 4.0 landscape. In *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 224–231. IEEE, 2019.

[94] Irlán Grangel-González, Paul Baptista, Lavdim Halilaj, Steffen Lohmann, Maria-Esther Vidal, Christian Mader, and Sören Auer. The industry 4.0 standards landscape from a semantic integration perspective. In *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–8. IEEE, 2017.

[95] Andreas Löcklin, Hannes Vietz, Dustin White, Tamás Ruppert, Nasser Jazdi, and Michael Weyrich. Data administration shell for data-science-driven development. *Procedia CIRP*, 100:115–120, 2021.

[96] Irlán Grangel-González, Lavdim Halilaj, Gökhan Coskun, Sören Auer, Diego Collarana, and Michael Hoffmeister. Towards a semantic administrative shell for industry 4.0 components. In *2016 IEEE Tenth International Conference on Semantic Computing (ICSC)*, pages 230–237. IEEE, 2016.

[97] Olga Kovalenko, Irlán Grangel-González, Marta Sabou, Arndt Lüder, Stefan Biffl, Sören Auer, and Maria-Esther Vidal. Automationml ontology: Modeling cyber-physical systems for industry 4.0. *IOS Press Journal*, 2018.

[98] Krzysztof Janowicz, Armin Haller, Simon JD Cox, Danh Le Phuoc, and Maxime Lefrançois. Sosa: A lightweight ontology for sensors, observations, samples, and actuators. *Journal of Web Semantics*, 56:1–10, 2019.

[99] Eeva Järvenpää, Niko Siltala, Otto Hylli, and Minna Lanz. The development of an ontology for describing the capabilities of manufacturing resources. *Journal of Intelligent Manufacturing*, 30(2):959–978, 2019.

[100] Nicolas Seydoux, Khalil Drira, Nathalie Hernandez, and Thierry Monteil. Iot-o, a core-domain iot ontology to represent connected devices networks. In *European Knowledge Acquisition Workshop*, pages 561–576. Springer, 2016.

[101] Robert Arp, Barry Smith, and Andrew D Spear. *Building ontologies with basic formal ontology.* Mit Press, 2015.

[102] JF Wang, JH Liu, and YF Zhong. A novel ant colony algorithm for assembly sequence planning. *The international journal of advanced manufacturing technology*, 25(11-12):1137–1143, 2005.

[103] Farhad Ameri and Debasish Dutta. An upper ontology for manufacturing service description. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 42578, pages 651–661, 2006.

[104] Aldo Gangemi, Nicola Guarino, Claudio Masolo, Alessandro Oltramari, and Luc Schneider. Sweetening ontologies with dolce. In *International Conference on Knowledge Engineering and Knowledge Management*, pages 166–181. Springer, 2002.

[105] Ansgar Scherp, Thomas Franz, Carsten Saathoff, and Steffen Staab. F–a model of events based on the foundational ontology dolce+ dns ultralight. In *Proceedings of the fifth international conference on Knowledge capture*, pages 137–144, 2009.

[106] Konstantinos Kotis and Artem Katasonov. An iot-ontology for the representation of interconnected, clustered and aligned smart entities. *Technical*

report, VTT Technical Research Center, Finland VTT Technical Research Center, Finland, 2012.

[107] Andrew Iliadis. The tower of babel problem: making data make sense with basic formal ontology. *Online Information Review*, 2019.

[108] International Organization for Standardization. Iso/iec 21838-2 information technology — top-level ontologies (tlo) — part 2: Basic formal ontology (bfo), 2021. `https://www.iso.org/standard/74572.html`.

[109] Yanru Zhong, Chaohao Jiang, Yuchu Qin, Guoyu Yang, Meifa Huang, and Xiaonan Luo. Automatically generating assembly sequences with an ontology-based approach. *Assembly Automation*, 2019.

[110] Fernando Roda and Estanislao Musulin. An ontology-based framework to support intelligent data analysis of sensor measurements. *Expert Systems with Applications*, 41(17):7914–7926, 2014.

[111] Valentina Presutti and Aldo Gangemi. Content ontology design patterns as practical building blocks for web ontologies. In *International Conference on Conceptual Modeling*, pages 128–141. Springer, 2008.

[112] Zhuoyu Huang, Casey Jowers, Ali Dehghan-Manshadi, and Matthew S Dargusch. Smart manufacturing and dvsm based on an ontological approach. *Computers in Industry*, 117:103189, 2020.

[113] Borja Ramis Ferrer, Bilal Ahmad, Andrei Lobov, Daniel Alexandre Vera, Jose Luis Martinez Lastra, and Robert Harrison. An approach for knowledge-driven product, process and resource mappings for assembly automation. In *2015 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 1104–1109. IEEE, 2015.

[114] Borja Ramis Ferrer, Wael M Mohammed, Andrei Lobov, Amalia Moreno Galera, and Jose L Martinez Lastra. Including human tasks as semantic resources in manufacturing ontology models. In *IECON 2017-43rd Annual Conference of the IEEE Industrial Electronics Society*, pages 3466–3473. IEEE, 2017.

[115] Anne-Francoise Cutting-Decelle, Robert IM Young, Jean-Jacques Michel, Reyes Grangel, Julie Le Cardinal, and Jean Pierre Bourey. Iso 15531 mandate: a product-process-resource based approach for managing modularity in production management. *Concurrent Engineering*, 15(2):217–235, 2007.

[116] Julius Pfrommer, Miriam Schleipen, and Jürgen Beyerer. Pprs: Production skills and their relation to product, process, and resource. In *2013 IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA)*, pages 1–4. IEEE, 2013.

[117] Mussawar Ahmad, Borja Ramis Ferrer, Bilal Ahmad, Daniel Vera, Jose L Martinez Lastra, and Robert Harrison. Knowledge-based ppr modelling for assembly automation. *CIRP Journal of Manufacturing Science and Technology*, 21:33–46, 2018.

[118] Oscar D Lara and Miguel A Labrador. A survey on human activity recognition using wearable sensors. *IEEE communications surveys & tutorials*, 15(3):1192–1209, 2012.

[119] Natalia Díaz Rodríguez, Manuel P Cuéllar, Johan Lilius, and Miguel Delgado Calvo-Flores. A survey on ontologies for human behavior recognition. *ACM Computing Surveys (CSUR)*, 46(4):1–33, 2014.

[120] Diane J Cook, Narayanan C Krishnan, and Parisa Rashidi. Activity discovery and activity recognition: A new partnership. *IEEE transactions on cybernetics*, 43(3):820–828, 2013.

[121] Juan Izquierdo-Domenech, Jordi Linares-Pellicer, and Jorge Orta-Lopez. Towards achieving a high degree of situational awareness and multimodal interaction with ar and semantic ai in industrial applications. *Multimedia Tools and Applications*, pages 1–27, 2022.

[122] Maryam Ziaeefard and Robert Bergevin. Semantic human activity recognition: A literature review. *Pattern Recognition*, 48(8):2329–2345, 2015.

[123] Mieczyslaw M Kokar, Christopher J Matheus, and Kenneth Baclawski. Ontology-based situation awareness. *Information fusion*, 10(1):83–98, 2009.

[124] Lucas Paletta, Amir Dini, Cornelia Murko, Saeed Yahyanejad, Michael Schwarz, Gerald Lodron, Stefan Ladstätter, Gerhard Paar, and Rosemarie Velik. Towards real-time probabilistic evaluation of situation awareness from

human gaze in human-robot interaction. In *Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, pages 247–248, 2017.

[125] Konstantinos Kotis and George A Vouros. Human-centered ontology engineering: The hcome methodology. *Knowledge and Information Systems*, 10(1):109–131, 2006.

[126] Cristina Aceta, Izaskun Fernández, and Aitor Soroa. Kide4i: A generic semantics-based task-oriented dialogue system for human-machine interaction in industry 5.0. *Applied Sciences*, 12(3):1192, 2022.

[127] Yi Zhou Jiang, Shi Fan Zhu, and Zhen Shang. An ontology-based framework for ergonomics knowledge classification and representation. In *Key Engineering Materials*, volume 572, pages 127–130. Trans Tech Publ, 2014.

[128] Wendy Elford. A multi-ontology view of ergonomics: Applying the cynefin framework to improve theory and practice. *Work*, 41(Supplement 1):812–817, 2012.

[129] Dave Snowden. The cynefin framework. *YouTube video*, 8:38, 2010.

[130] Fazel Ansari, Marjan Khobreh, Ulrich Seidenberg, and Wilfried Sihn. A problem-solving ontology for human-centered cyber physical production systems. *CIRP Journal of Manufacturing Science and Technology*, 22:91–106, 2018.

[131] International standard iso 22400-1. automation systems and integration—key performance indicators (kpis) for manufacturing operations management—part 1: Overview, concepts and terminology. Standard, Organization (ISO), 2014.

[132] Borja Ramis Ferrer, Usman Muhammad, Wael M Mohammed, and José L Martínez Lastra. Implementing and visualizing iso 22400 key performance indicators for monitoring discrete manufacturing systems. *Machines*, 6(3):39, 2018.

[133] Li Zhu, Charlotta Johnsson, Martina Varisco, and Massimiliano M Schiraldi. Key performance indicators for manufacturing operations management–gap analysis between process industrial needs and iso 22400 standard. *Procedia Manufacturing*, 25:82–88, 2018.

[134] Iiro Harjunkoski, Martin Hollender, Reinhard Bauer, Jens Doppelhamer, Subanatarajan Subbiah, and Werner Schmidt. An easy-to-use and flexible scheduling component. *Control and productivity*, 2017.

[135] Seyedamir Ahmadi. Ontology validation of manufacturing execution systems through the analysis of semantic descriptions - master's thesis, 2018.

[136] Irlán Grangel-González, Lavdim Halilaj, Maria-Esther Vidal, Omar Rana, Steffen Lohmann, Sören Auer, and Andreas W Müller. Knowledge graphs for semantically integrating cyber-physical systems. In *International Conference on Database and Expert Systems Applications*, pages 184–199. Springer, 2018.

[137] Greyce N Schroeder, Charles Steinmetz, Carlos E Pereira, and Danubia B Espindola. Digital twin data modeling with automationml and a communication methodology for data exchange. *IFAC-PapersOnLine*, 49(30):12–17, 2016.

[138] Bernhard Wally, Laurens Lang, Rafał Włodarski, Radek Šindelár, Christian Huemer, Alexandra Mazak, and Manuel Wimmer. Generating structured automationml models from iec 62264 information. *Proceedings of the 5th AutomationML PlugFest 2019*, 2019.

[139] Angelos Argyrou, Christos Giannoulis, Nikolaos Papakostas, and George Chryssolouris. A uniform data model for representing symbiotic assembly stations. *Procedia CIRP*, 44:85–90, 2016.

[140] Rainer Müller, Leenhard Hörauf, Matthias Vette, and Christoph Speicher. Planning and developing cyber-physical assembly systems by connecting virtual and real worlds. *Procedia Cirp*, 52:35–40, 2016.

[141] Ralf Kretschmer, Alain Pfouga, Stefan Rulhoff, and J Stjepandić. Knowledge-based design for assembly in agile manufacturing by using data mining methods. *Advanced Engineering Informatics*, 33:285–299, 2017.

[142] Martin Ringsquandl, Steffen Lamparter, Sebastian Brandt, Thomas Hubauer, and Raffaello Lepratti. Semantic-guided feature selection for industrial automation systems. In *International Semantic Web Conference*, pages 225–240. Springer, 2015.

[143] Evgeny Kharlamov, Ernesto Jiménez-Ruiz, Dmitriy Zheleznyakov, Dimitris Bilidas, Martin Giese, Peter Haase, Ian Horrocks, Herald Kllapi, Manolis Koubarakis, Özgür Özçep, et al. Optique: Towards obda systems for industry. In *Extended Semantic Web Conference*, pages 125–140. Springer, 2013.

[144] Evgeny Kharlamov, Nina Solomakhina, Özgür Lütfü Özçep, Dmitriy Zheleznyakov, Thomas Hubauer, Steffen Lamparter, Mikhail Roshchin, Ahmet Soylu, and Stuart Watson. How semantic technologies can enhance data access at siemens energy. In *International Semantic Web Conference*, pages 601–619. Springer, 2014.

[145] Evgeny Kharlamov, Bernardo Cuenca Grau, Ernesto Jiménez-Ruiz, Steffen Lamparter, Gulnar Mehdi, Martin Ringsquandl, Yavor Nenov, Stephan Grimm, Mikhail Roshchin, and Ian Horrocks. Capturing industrial information models with ontologies and constraints. In *International Semantic Web Conference*, pages 325–343. Springer, 2016.

[146] Martin Ringsquandl, Steffen Lamparter, and Raffaello Lepratti. Graph-based predictions and recommendations in flexible manufacturing systems. In *IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society*, pages 6937–6942. IEEE, 2016.

[147] Jonathan Hayes and Claudio Gutierrez. Bipartite graphs as intermediate model for rdf. In *International Semantic Web Conference*, pages 47–61. Springer, 2004.

[148] Gergely Honti and János Abonyi. Frequent itemset mining and multi-layer network-based analysis of rdf databases. *Mathematics*, 9(4):450, 2021.

[149] Bin Zhou, Jinsong Bao, Jie Li, Yuqian Lu, Tianyuan Liu, and Qiwan Zhang. A novel knowledge graph-based optimization approach for resource allocation in discrete manufacturing workshops. *Robotics and Computer-Integrated Manufacturing*, 71:102160, 2021.

[150] László Grad-Gyenge, Attila Kiss, and Peter Filzmoser. Graph embedding based recommendation techniques on the knowledge graph. In *Adjunct publication of the 25th conference on user modeling, adaptation and personalization*, pages 354–359, 2017.

[151] Gergő Gombos and Attila Kiss. Sparql query writing with recommendations based on datasets. In *International Conference on Human Interface and the Management of Information*, pages 310–319. Springer, 2014.

[152] Gunilla Sivard, Navid Shariatzadeh, and Hakan Akillioglu. Modules information modelling in evolvable production systems. In *5th Swedish Production Symposium, SPS12*, 2012.

[153] Mohammed M Mabkhot, Sana Kouki Amri, Saber Darmoul, Ali M Al-Samhan, and Sabeur Elkosantini. An ontology-based multi-criteria decision support system to reconfigure manufacturing systems. *IISE Transactions*, 52(1):18–42, 2020.

[154] Marvin Carl May, Lars Kiefer, Andreas Kuhnle, and Gisela Lanza. Ontology-based production simulation with ontologysim. *Applied Sciences*, 12(3):1608, 2022.

[155] Wiking Jurasky, Patrick Moder, Michael Milde, Hans Ehm, and Gunther Reinhart. Transformation of semantic knowledge into simulation-based decision support. *Robotics and Computer-Integrated Manufacturing*, 71:102174, 2021.

[156] Shiyong Wang, Jiafu Wan, Di Li, and Chengliang Liu. Knowledge reasoning with semantic data for real-time data processing in smart factory. *Sensors*, 18(2):471, 2018.

[157] Zhenyu Wu, Yuan Xu, Yunong Yang, Chunhong Zhang, Xinning Zhu, and Yang Ji. Towards a semantic web of things: a hybrid semantic annotation, extraction, and reasoning framework for cyber-physical system. *Sensors*, 17(2):403, 2017.

[158] Evgeny Kharlamov, Gulnar Mehdi, Ognjen Savković, Guohui Xiao, Elem Güzel Kalaycı, and Mikhail Roshchin. Semantically-enhanced rule-based diagnostics for industrial internet of things: The sdrl language and case study for siemens trains and turbines. *Journal of web semantics*, 56:11–29, 2019.

[159] Borja Ramis Ferrer, Wael M Mohammed, Mussawar Ahmad, Sergii Iarovyi, Jiayi Zhang, Robert Harrison, and Jose Luis Martinez Lastra. Comparing ontologies and databases: a critical review of lifecycle engineering models

in manufacturing. *Knowledge and Information Systems*, 63(6):1271–1304, 2021.

[160] Sai Sumathi and S Esakkirajan. *Fundamentals of relational database management systems*, volume 47. Springer, 2007.

[161] Cinzia Daraio, Maurizio Lenzerini, Claudio Leporelli, Paolo Naggar, Andrea Bonaccorsi, and Alessandro Bartolucci. The advantages of an ontology-based data management approach: Openness, interoperability and data quality. *Scientometrics*, 108:441–455, 2016.

[162] Carmen Martinez-Cruz, Ignacio J Blanco, and M Amparo Vila. Ontologies versus relational databases: are they so different? a comparison. *Artificial Intelligence Review*, 38:271–290, 2012.

[163] Artem Chebotko, Shiyong Lu, and Farshad Fotouhi. Semantics preserving sparql-to-sql translation. *Data & Knowledge Engineering*, 68(10):973–1000, 2009.

[164] Mariano Rodriguez-Muro and Martin Rezk. Efficient sparql-to-sql with r2rml mappings. *Journal of Web Semantics*, 33:141–169, 2015.

[165] Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29, 2000.

[166] Hans-Jörg Happel and Stefan Seedorf. Applications of ontologies in software engineering. In *Proc. of Workshop on Sematic Web Enabled Software Engineering"(SWESE) on the ISWC*, pages 5–9. Citeseer, 2006.

[167] NS Ong and G Boothroyd. Assembly times for electrical connections and wire harnesses. *The International Journal of Advanced Manufacturing Technology*, 6(2):155–179, 1991.

[168] Stanford University. Protégé - a free, open-source ontology editor and framework for building intelligent systems, 2020. https://protege.stanford.edu/.

[169] Ontotext. Graphdb - an enterprise ready semantic graph database, compliant with w3c standards., 2020. https://graphdb.ontotext.com/.

[170] Natalya Fridman Noy, Monica Crubézy, Ray W Fergerson, Holger Knublauch, Samson W Tu, Jennifer Vendetti, and Mark A Musen. Protégé-2000: an open-source ontology-development and knowledge-acquisition environment. In *AMIA... Annual Symposium proceedings. AMIA Symposium*, pages 953–953, 2003.

[171] John H Gennari, Mark A Musen, Ray W Fergerson, William E Grosso, Monica Crubézy, Henrik Eriksson, Natalya F Noy, and Samson W Tu. The evolution of protégé: an environment for knowledge-based systems development. *International Journal of Human-computer studies*, 58(1):89–123, 2003.

[172] Pierre-Yves Vandenbussche, Ghislain A Atemezing, María Poveda-Villalón, and Bernard Vatant. Linked open vocabularies (lov): a gateway to reusable semantic vocabularies on the web. *Semantic Web*, 8(3):437–452, 2017.

[173] Haibo Cheng, Peng Zeng, Lingling Xue, Zhao Shi, Peng Wang, and Haibin Yu. Manufacturing ontology development based on industry 4.0 demonstration production line. In *2016 Third International Conference on Trustworthy Systems and their Applications (TSA)*, pages 42–47. IEEE, 2016.

[174] Steffen Lohmann, Stefan Negru, and David Bold. The protégévowl plugin: ontology visualization for everyone. In *European Semantic Web Conference*, pages 395–400. Springer, 2014.

[175] Niklas Petersen Irlan Grangel. Rami vocabulary, 2015. https://i40.semantic-interoperability.org/rami/Documentation/index.html.

[176] Niklas Petersen Christoph Lange, Irlan Grangel. Semantic manufacturing ontology, 2018. https://i40.semantic-interoperability.org/smo/.

[177] Deborah McGuinness Rensselaer Timothy Lebo, Satya Sahoo. Prov-o: The prov ontology, 2013. `https://www.w3.org/TR/prov-o/`.

[178] Sebastian Tramp Irlan Grangel. Scorvoc vocabulary, 2015. `https://github.com/vocol/scor`.

[179] Aldo Gangemi. The dolce+dns ultralite ontology, 2010. `http://ontologydesignpatterns.org/wiki/Ontology:DOLCE+DnS_Ultralite`.

[180] Dejing Dou, Hao Wang, and Haishan 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)*, pages 244–251. IEEE, 2015.

[181] Stephen P Gardner. Ontologies and semantic data integration. *Drug discovery today*, 10(14):1001–1007, 2005.

[182] Margherita Peruzzini, Fabio Grandi, and Marcello Pellicciari. Exploring the potential of operator 4.0 interface and monitoring. *Computers & Industrial Engineering*, 139:105600, 2020.

[183] Mohd Javaid and Abid Haleem. Critical components of industry 5.0 towards a successful adoption in the field of manufacturing. *Journal of Industrial Integration and Management*, 5(03):327–348, 2020.

[184] Charlotta Johnsson Dennis Brandl. Beyond the pyramid: Using isa95 for industry 4.0 and smart manufacturing, 2022. `https://www.automation.com/en-us/articles/january-2022/beyond-pyramid-isa95-industry-4-0`.

[185] Lei Yue, Pengfei Niu, and Yan Wang. Guidelines for defining user requirement specifications (urs) of manufacturing execution system (mes) based on isa-95 standard. In *Journal of Physics: Conference Series*, volume 1168, page 032065. IOP Publishing, 2019.

[186] Alican Dogan and Derya Birant. Machine learning and data mining in manufacturing. *Expert Systems with Applications*, 166:114060, 2021.

[187] Thorsten Wuest, Daniel Weimer, Christopher Irgens, and Klaus-Dieter Thoben. Machine learning in manufacturing: advantages, challenges, and applications. *Production & Manufacturing Research*, 4(1):23–45, 2016.

[188] Jorge F Arinez, Qing Chang, Robert X Gao, Chengying Xu, and Jianjing Zhang. Artificial intelligence in advanced manufacturing: Current status and future outlook. *Journal of Manufacturing Science and Engineering*, 142(11), 2020.

[189] Douglas H Rothenberg. *Alarm management for process control: a best-practice guide for design, implementation, and use of industrial alarm systems.* Momentum Press, 2009.

[190] Pai Zheng, Zhiqian Sang, Ray Y Zhong, Yongkui Liu, Chao Liu, Khamdi Mubarok, Shiqiang Yu, Xun Xu, et al. Smart manufacturing systems for industry 4.0: Conceptual framework, scenarios, and future perspectives. *Frontiers of Mechanical Engineering*, 13(2):137–150, 2018.

[191] Zexuan Zhu, Chao Liu, and Xun Xu. Visualisation of the digital twin data in manufacturing by using augmented reality. *Procedia Cirp*, 81:898–903, 2019.

[192] Payal Thakur and Vivek Kumar Sehgal. Emerging architecture for heterogeneous smart cyber-physical systems for industry 5.0. *Computers & Industrial Engineering*, 162:107750, 2021.

[193] Pankaj Pathak, Parashu Ram Pal, Manish Shrivastava, and Priyanka Ora. Fifth revolution: Applied ai & human intelligence with cyber physical systems. *International Journal of Engineering and Advanced Technology*, 8(3):23–27, 2019.

[194] Patrik Grznár, Milan Gregor, Martin Krajčovič, Štefan Mozol, Marek Schickerle, Vladimír Vavrík, Lukáš Ďurica, Martin Marschall, and Tomáš Bielik. Modeling and simulation of processes in a factory of the future. *Applied sciences*, 10(13):4503, 2020.

[195] Svenja Korder, Barbara Tropschuh, and Gunther Reinhart. A competence-based description of employees in reconfigurable manufacturing systems. In *IFIP International Conference on Advances in Production Management Systems*, pages 257–264. Springer, 2019.

[196] Fazel Ansari, Philipp Hold, and Marjan Khobreh. A knowledge-based approach for representing jobholder profile toward optimal human–machine collaboration in cyber physical production systems. *CIRP Journal of manufacturing science and technology*, 28:87–106, 2020.

[197] Reynold Cheng and Sunil Prabhakar. Managing uncertainty in sensor database. *ACM SIGMOD Record*, 32(4):41–46, 2003.

[198] Jeremy A Marvel, Shelly Bagchi, Megan Zimmerman, and Brian Antonishek. Towards effective interface designs for collaborative hri in manufacturing: metrics and measures. *ACM Transactions on Human-Robot Interaction (THRI)*, 9(4):1–55, 2020.

[199] Evert Helms, Rolf Dieter Schraft, and M Hagele. rob@ work: Robot assistant in industrial environments. In *Proceedings. 11th IEEE International Workshop on Robot and Human Interactive Communication*, pages 399–404. IEEE, 2002.

[200] Adrien Malaisé, Pauline Maurice, Francis Colas, and Serena Ivaldi. Activity recognition for ergonomics assessment of industrial tasks with automatic feature selection. *IEEE Robotics and Automation Letters*, 4(2):1132–1139, 2019.

[201] Aaron Steinfeld, Terrence Fong, David Kaber, Michael Lewis, Jean Scholtz, Alan Schultz, and Michael Goodrich. Common metrics for human-robot interaction. In *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, pages 33–40, 2006.

[202] Amon Göppert, Lea Grahn, Jonas Rachner, Dennis Grunert, Simon Hort, and Robert H Schmitt. Pipeline for ontology-based modeling and automated deployment of digital twins for planning and control of manufacturing systems. *Journal of Intelligent Manufacturing*, pages 1–20, 2021.

[203] Giancarlo Perrone. Pyvis interactive network visualizations, 2018.

[204] Paco Nathan. kglab: a simple abstraction layer in Python for building knowledge graphs, 2020.

[205] Paco Nathan. Graph-based approaches for hybrid ai solutions. 2021.

[206] Bob DuCharme. *Learning SPARQL: querying and updating with SPARQL 1.1.* " O'Reilly Media, Inc.", 2013.

[207] Muhammad Yahya, John G Breslin, and Muhammad Intizar Ali. Semantic web and knowledge graphs for industry 4.0. *Applied Sciences*, 11(11):5110, 2021.

[208] Armin Haller, Krzysztof Janowicz, Simon JD Cox, Maxime Lefrançois, Kerry Taylor, Danh Le Phuoc, Joshua Lieberman, Raúl García-Castro, Rob Atkinson, and Claus Stadler. The modular ssn ontology: A joint w3c and ogc standard specifying the semantics of sensors, observations, sampling, and actuation. *Semantic Web*, 10(1):9–32, 2019.

[209] Albert-László Barabási. Network science. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1987):20120375, 2013.

[210] Wei Zhang, Xiaogang Wang, Deli Zhao, and Xiaoou Tang. Graph degree linkage: Agglomerative clustering on a directed graph. In *European conference on computer vision*, pages 428–441. Springer, 2012.

[211] John A Barnes and Frank Harary. Graph theory in network analysis. *Social networks*, 5(2):235–244, 1983.

[212] Oded Tal. Overall resources effectiveness, the key for cycle time reduction & capacity improvements. In *GaAs Mantech Conference*, pages 255–8, 2001.

[213] Mohammed Badawy, AA Abd El-Aziz, Amira M Idress, Hesham Hefny, and Shrouk Hossam. A survey on exploring key performance indicators. *Future Computing and Informatics Journal*, 1(1-2):47–52, 2016.

[214] Stefano Boccaletti, Ginestra Bianconi, Regino Criado, Charo I Del Genio, Jesús Gómez-Gardenes, Miguel Romance, Irene Sendina-Nadal, Zhen Wang, and Massimiliano Zanin. The structure and dynamics of multilayer networks. *Physics Reports*, 544(1):1–122, 2014.

[215] Mikko Kivelä, Alex Arenas, Marc Barthelemy, James P Gleeson, Yamir Moreno, and Mason A Porter. Multilayer networks. *Journal of Complex Networks*, 2(3):203–271, 2014.

[216] Tamás Ruppert, Gergely Honti, and János Abonyi. Multilayer network-based production flow analysis. *Complexity*, 2018, 2018.

[217] Csaba Pigler, Ágnes Fogarassy-Vathy, and János Abonyi. Scalable co-clustering using a crossing minimization–application to production flow analysis. *Acta Polytechnica Hungarica*, 13(2):209–228, 2016.

[218] Christian Becker and Armin Scholl. A survey on problems and methods in generalized assembly line balancing. *European Journal of Operational Research*, 168(3):694–715, 2006.

[219] Nils Boysen, Malte Fliedner, and Armin Scholl. Assembly line balancing: Which model to use when? *International Journal of Production Economics*, 111(2):509–528, 2008.

[220] J Mukund Nilakantan, SG Ponnambalam, and Peter Nielsen. Application of particle swarm optimization to solve robotic assembly line balancing problems. In *Handbook of Neural Computation*, pages 239–267. Elsevier, 2017.

[221] Öncü Hazır, Xavier Delorme, and Alexandre Dolgui. A survey on cost and profit oriented assembly line balancing. *IFAC Proceedings Volumes*, 47(3):6159–6167, 2014.

[222] H Edwin Romeijn and Robert L Smith. Simulated annealing for constrained global optimization. *Journal of Global Optimization*, 5(2):101–126, 1994.

[223] G Suresh and S Sahu. Stochastic assembly line balancing using simulated annealing. *The International Journal of Production Research*, 32(8):1801–1810, 1994.

[224] Wen-Chyuan Chiang and Timothy L Urban. The stochastic u-line balancing problem: A heuristic procedure. *European Journal of Operational Research*, 175(3):1767–1781, 2006.

[225] Kürşad Ağpak and Hadi Gökçen. A chance-constrained approach to stochastic line balancing problem. *European Journal of Operational Research*, 180(3):1098–1115, 2007.

[226] Franco Guerriero and John Miltenburg. The stochastic u-line balancing problem. *Naval Research Logistics (NRL)*, 50(1):31–57, 2003.

[227] Wen-Chyuan Chiang. The application of a tabu search metaheuristic to the assembly line balancing problem. *Annals of Operations Research*, 77:209–227, 1998.

[228] Yasuhiro Tsujimura, Mitsuo Gen, and Erika Kubota. Solving fuzzy assembly-line balancing problem with genetic algorithms. *Computers & Industrial Engineering*, 29(1-4):543–547, 1995.

[229] Uğur Özcan and Bilal Toklu. Multiple-criteria decision-making in two-sided assembly line balancing: A goal programming and a fuzzy goal programming models. *Computers & Operations Research*, 36(6):1955–1965, 2009.

[230] Adil Baykasoglu. Multi-rule multi-objective simulated annealing algorithm for straight and u type assembly line balancing problems. *Journal of Intelligent Manufacturing*, 17(2):217–232, 2006.

[231] Patrick R McMullen and GV Frazier. Using simulated annealing to solve a multiobjective assembly line balancing problem with parallel workstations. *International Journal of Production Research*, 36(10):2717–2741, 1998.

[232] H Güden and S Meral. An adaptive simulated annealing algorithm-based approach for assembly line balancing and a real-life case study. *The International Journal of Advanced Manufacturing Technology*, 84(5-8):1539–1559, 2016.

[233] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.

[234] Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. Benchmark graphs for testing community detection algorithms. *Physical review E*, 78(4):046110, 2008.

[235] Jaewon Yang, Julian McAuley, and Jure Leskovec. Community detection in networks with node attributes. In *2013 IEEE 13th International Conference on Data Mining*, pages 1151–1156. IEEE, 2013.

[236] Jörg Reichardt and Stefan Bornholdt. Statistical mechanics of community detection. *Physical review E*, 74(1):016110, 2006.

[237] Wenjian Luo, Nannan Lu, Li Ni, Wenjie Zhu, and Weiping Ding. Local community detection by the nearest nodes with greater centrality. *Information Sciences*, 517:377–392, 2020.

[238] Muhammad Aqib Javed, Muhammad Shahzad Younis, Siddique Latif, Junaid Qadir, and Adeel Baig. Community detection in networks: A multidisciplinary review. *Journal of Network and Computer Applications*, 108:87–111, 2018.

[239] Sayan Ghosh, Mahantesh Halappanavar, Antonino Tumeo, Ananth Kalyanaraman, Hao Lu, Daniel Chavarria-Miranda, Arif Khan, and Assefaw Gebremedhin. Distributed louvain algorithm for graph community detection. In *2018 IEEE international parallel and distributed processing symposium (IPDPS)*, pages 885–895. IEEE, 2018.

[240] Yansen Su, Kefei Zhou, Xingyi Zhang, Ran Cheng, and Chunhou Zheng. A parallel multi-objective evolutionary algorithm for community detection in large-scale complex networks. *Information Sciences*, 576:374–392, 2021.

[241] Dongxiao He, Youyou Wang, Jinxin Cao, Weiping Ding, Shizhan Chen, Zhiyong Feng, Bo Wang, and Yuxiao Huang. A network embedding-enhanced bayesian model for generalized community detection in complex networks. *Information Sciences*, 575:306–322, 2021.

[242] Aaron Clauset, Mark EJ Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical review E*, 70(6):066111, 2004.

[243] Xingwang Zhao, Jiye Liang, and Jie Wang. A community detection algorithm based on graph compression for large-scale social networks. *Information Sciences*, 551:358–372, 2021.

[244] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[245] Jinrong Zhu, Bilian Chen, and Yifeng Zeng. Community detection based on modularity and k-plexes. *Information Sciences*, 513:127–142, 2020.

[246] Andrea Lancichinetti and Santo Fortunato. Community detection algorithms: a comparative analysis. *Physical review E*, 80(5):056117, 2009.

[247] Santo Fortunato and Darko Hric. Community detection in networks: A user guide. *Physics reports*, 659:1–44, 2016.

[248] Santo Fortunato. Community detection in graphs. *Physics reports*, 486(3-5):75–174, 2010.

[249] Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582, 2006.

[250] Zhan Bu, Chengcui Zhang, Zhengyou Xia, and Jiandong Wang. A fast parallel modularity optimization algorithm (fpmqa) for community detection in online social network. *Knowledge-Based Systems*, 50:246–259, 2013.

[251] Zhishan Dong, Shuangshuang Wang, and Qun Liu. Spectral based hypothesis testing for community detection in complex networks. *Information Sciences*, 512:1360–1371, 2020.

[252] Alain Bretto. Hypergraph theory. *An introduction. Mathematical Engineering. Cham: Springer*, 2013.

[253] Claude Berge. Graphs and hypergraphs. 1973.

[254] E Vasilyeva, A Kozlov, K Alfaro-Bittner, D Musatov, AM Raigorodskii, M Perc, and S Boccaletti. Multilayer representation of collaboration networks with higher-order interactions. *Scientific reports*, 11(1):1–11, 2021.

[255] Dengyong Zhou, Jiayuan Huang, and Bernhard Schölkopf. Learning with hypergraphs: Clustering, classification, and embedding. *Advances in neural information processing systems*, 19, 2006.

[256] Guang Zhang. Adaptive allocation rules for hypergraph games. *Operations Research Letters*, 49(6):890–895, 2021.

[257] David A Kosian and Leon A Petrosyan. Two-level cooperative game on hypergraph. *Contributions to Game Theory and Management*, 14(0):227–235, 2021.

[258] Yunpeng Bai, Chen Gong, Bin Zhang, Guoliang Fan, and Xinwen Hou. Value function factorisation with hypergraph convolution for cooperative multi-agent reinforcement learning. *arXiv preprint arXiv:2112.06771*, 2021.

[259] Enea Scioni, Nico Hübel, Sebastian Blumenthal, Azamat Shakhimardanov, Markus Klotzbücher, Hugo Garcia, and Herman Bruyninckx. Hierarchical hypergraph for knowledge-centric robot systems: a composable structural meta model and its domain specific language npc4. *JOSER: Journal of Software Engineering for Robotics*, 7(11):55–74, 2016.

[260] Lei Wu, Minggang He, and Yanyan Han. Hypergraph clustering-based cloud manufacturing service management method. In *Proceedings of the 2014 IEEE 18th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 220–225. IEEE, 2014.

[261] Housni Djellab, Khaled Djellab, and Michel Gourgand. A new heuristic based on a hypergraph representation for the tool switching problem. *International Journal of Production Economics*, 64(1-3):165–176, 2000.

[262] Liqiao Xia, Pai Zheng, Xiao Huang, and Chao Liu. A novel hypergraph convolution network-based approach for predicting the material removal rate in chemical mechanical planarization. *Journal of Intelligent Manufacturing*, pages 1–12, 2021.

[263] IR Aliyev, JF Mammadov, and Sh R Rakhimov. Selection of information-measuring components on the basis of layout diagram of flexible manufacturing cell. In *International Conference on Theory and Application of Soft Computing, Computing with Words and Perceptions*, pages 124–131. Springer, 2021.

[264] Nestor F Michelena and Panos Y Papalambros. A hypergraph framework for optimal model-based decomposition of design problems. *Computational optimization and applications*, 8(2):173–196, 1997.

[265] Levent Kandiller. A cell formation algorithm: Hypergraph approximation-cut tree. *European journal of operational research*, 109(3):686–702, 1998.

[266] Levent Kandiller. A combinatorial optimization tour in cell formation via hypergraphs. *Group Technology And Cellular Manufacturing: Methodologies and Applications*, page 55, 2020.

[267] Lantao Liu and Dylan A Shell. Multi-level partitioning and distribution of the assignment problem for large-scale multi-robot task allocation. *Robotics: Science and Systems VII; MIT Press: Cambridge, MA, USA*, pages 26–33, 2011.

[268] Zixiang Nie and Kwang-Cheng Chen. Hypergraphical real-time multi-robot task allocation in a smart factory. *IEEE Transactions on Industrial Informatics*, 2021.

[269] Antonio P Volpentesta and Alberto M Felicetti. Competence mapping through analysing research papers of a scientific community. In *Doctoral Conference on Computing, Electrical and Industrial Systems*, pages 33–44. Springer, 2011.

[270] Piet de Jong. A statistical approach to saaty's scaling method for priorities. *Journal of Mathematical Psychology*, 28(4):467–478, 1984.

[271] William Ho and Xin Ma. The state-of-the-art integrations and applications of the analytic hierarchy process. *European Journal of Operational Research*, 267(2):399–414, 2018.

[272] Thomas L Saaty. Decision making—the analytic hierarchy and network processes (AHP/ANP). *Journal of Systems Science and Systems Engineering*, 13(1):1–35, 2004.

[273] Patrycjusz Stoma, Monika Stoma, Agnieszka Dudziak, and Jacek Caban. Bootstrap analysis of the production processes capability assessment. *Applied Sciences*, 9(24):5360, 2019.

[274] ZH Che. Clustering and selecting suppliers based on simulated annealing algorithms. *Computers & Mathematics with Applications*, 63(1):228–238, 2012.

[275] Elizabeth Dodson Coulter, John Sessions, and Michael G Wing. Scheduling forest road maintenance using the analytic hierarchy process and heuristics. *Silva Fennica*, 40(1):143–160, 2006.

[276] Yan Lu, Katherine C Morris, and Simon Frechette. Current standards landscape for smart manufacturing systems. *National Institute of Standards and Technology, NISTIR*, 8107:39, 2016.

[277] Benjamin Bryton. *Balancing of a continuous production line.* PhD thesis, Northwestern University, 1954.

[278] Erdal Erel and Subhash C Sarin. A survey of the assembly line balancing procedures. *Production Planning & Control*, 9(5):414–434, 1998.

[279] Daniel Leitold, Agnes Vathy-Fogarassy, and Janos Abonyi. Empirical working time distribution-based line balancing with integrated simulated annealing and dynamic programming. *Central European Journal of Operations Research*, 27(2):455–473, 2019.

[280] Steven T Hackman, Michael J Magazine, and TS Wee. Fast, effective algorithms for simple assembly line balancing problems. *Operations Research*, 37(6):916–924, 1989.

[281] Armin Scholl and Stefan Voß. Simple assembly line balancing - heuristic approaches. *Journal of Heuristics*, 2(3):217–244, 1997.

[282] Thomas R Hoffmann. Assembly line balancing with a precedence matrix. *Management Science*, 9(4):551–562, 1963.

[283] Earl D Sacerdoti. A structure for plans and behavior. Technical report, SRI International, Menlo Park, CA, Artificial Intelligence Center, 1975.

[284] James R Jackson. A computing procedure for a line balancing problem. *Management Science*, 2(3):261–271, 1956.

[285] Tamas Ruppert and Janos Abonyi. Software sensor for activity-time monitoring and fault detection in production lines. *Sensors*, 18(7):2346, 2018.

[286] Manlio De Domenico. Muxviz - the multilayer analysis and visualization platform., 2020. https://muxviz.net/.

[287] Scott Kirkpatrick, C Daniel Gelatt Jr, and Mario P Vecchi. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983.

[288] M Emin Aydin and Terence C Fogarty. A distributed evolutionary simulated annealing algorithm for combinatorial optimisation problems. *Journal of heuristics*, 10:269–292, 2004.

[289] Xinyu Que, Fabio Checconi, Fabrizio Petrini, and John A Gunnels. Scalable community detection with the louvain algorithm. In *2015 IEEE International Parallel and Distributed Processing Symposium*, pages 28–37. IEEE, 2015.

[290] Pasquale De Meo, Emilio Ferrara, Giacomo Fiumara, and Alessandro Provetti. Generalized louvain method for community detection in large networks. In *2011 11th international conference on intelligent systems design and applications*, pages 88–93. IEEE, 2011.

[291] Albert-László Barabási et al. *Network science*. Cambridge university press, 2016.

[292] Ulrik Brandes, Daniel Delling, Marco Gaertler, Robert Gorke, Martin Hoefer, Zoran Nikoloski, and Dorothea Wagner. On modularity clustering. *IEEE transactions on knowledge and data engineering*, 20(2):172–188, 2007.

[293] Christian Unkelbach, Klaus Fiedler, Myriam Bayer, Martin Stegmüller, and Daniel Danner. Why positive information is processed faster: The density hypothesis. *Journal of personality and social psychology*, 95(1):36, 2008.

[294] Nicolas Dugué and Anthony Perez. *Directed Louvain: maximizing modularity in directed networks*. PhD thesis, Université d'Orléans, 2015.

[295] Jeongah Yoon, Anselm Blumer, and Kyongbum Lee. An algorithm for modularity analysis of directed and weighted biological networks based on edge-betweenness centrality. *Bioinformatics*, 22(24):3106–3108, 2006.

[296] Santo Fortunato and Marc Barthelemy. Resolution limit in community detection. *Proceedings of the national academy of sciences*, 104(1):36–41, 2007.

[297] Waseem Ahmad and Ashfaq Khokhar. chawk: an efficient biclustering algorithm based on bipartite graph crossing minimization. In *VLDB Workshop on Data Mining in Bioinformatics*, pages 1–12. Citeseer, 2007.

[298] Ahsan Abdullah and Amir Hussain. A new biclustering technique based on crossing minimization. *Neurocomputing*, 69(16-18):1882–1896, 2006.

[299] Jon M Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, 1999.

[300] Mitri Kitti. Axioms for centrality scoring with principal eigenvectors. *Social choice and welfare*, 46(3):639–653, 2016.

[301] Nicola Perra and Santo Fortunato. Spectral centrality measures in complex networks. *Physical Review E*, 78(3):036107, 2008.

[302] Laishui Lv, Kun Zhang, Ting Zhang, Xun Li, Jiahui Zhang, and Wei Xue. Eigenvector centrality measure based on node similarity for multilayer and temporal networks. *IEEE Access*, 7:115725–115733, 2019.

[303] Jerrold H Zar. Spearman rank correlation. *Encyclopedia of biostatistics*, 7, 2005.

[304] Eamonn Keogh, Selina Chu, David Hart, and Michael Pazzani. An online algorithm for segmenting time series. In *Proceedings 2001 IEEE international conference on data mining*, pages 289–296. IEEE, 2001.

[305] László Dobos and János Abonyi. Fisher information matrix based time-series segmentation of process data. *Chemical Engineering Science*, 101:99–108, 2013.

[306] Sanja Fidler, Roozbeh Mottaghi, Alan Yuille, and Raquel Urtasun. Bottom-up segmentation for top-down detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3294–3301, 2013.

[307] Eran Borenstein and Shimon Ullman. Combined top-down/bottom-up segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 30(12):2109–2125, 2008.

[308] Johan Himberg, Kalle Korpiaho, Heikki Mannila, Johanna Tikanmaki, and Hannu TT Toivonen. Time series segmentation for context recognition in mobile devices. In *Proceedings 2001 IEEE International Conference on Data Mining*, pages 203–210. IEEE, 2001.

[309] Michele Coscia. The atlas for the aspiring network scientist. *arXiv preprint arXiv:2101.00863*, 2021.

[310] Jussi M Kumpula, Jari Saramäki, Kimmo Kaski, and Janos Kertesz. Limited resolution and multiresolution methods in complex network community detection. *Fluctuation and Noise Letters*, 7(03):L209–L214, 2007.

[311] Tapio Heimo, Jussi M Kumpula, Kimmo Kaski, and Jari Saramäki. Detecting modules in dense weighted networks with the potts method. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(08):P08007, 2008.

[312] Jaewon Yang and Jure Leskovec. Defining and evaluating network communities based on ground-truth. In *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*, pages 1–8, 2012.

[313] Mukkai S Krishnamoorthy. An np-hard problem in bipartite graphs. *ACM SIGACT News*, 7(1):26–26, 1975.

[314] Juris Hartmanis and Richard E Stearns. On the computational complexity of algorithms. *Transactions of the American Mathematical Society*, 117:285–306, 1965.

[315] Leon Danon, Albert Diaz-Guilera, Jordi Duch, and Alex Arenas. Comparing community structure identification. *Journal of statistical mechanics: Theory and experiment*, 2005(09):P09008, 2005.

[316] Erzsébet Ravasz, Anna Lisa Somera, Dale A Mongru, Zoltán N Oltvai, and A-L Barabási. Hierarchical organization of modularity in metabolic networks. *science*, 297(5586):1551–1555, 2002.

[317] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.

[318] Christian Toth. A collection of lfr benchmark graphs, January 2021.

[319] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. `http://snap.stanford.edu/data`, June 2014.

[320] Ryan Rossi and Nesreen Ahmed. The network data repository with interactive graph analytics and visualization. In *Twenty-ninth AAAI conference on artificial intelligence*, pages 4292–4293, 2015.

[321] William H Press, Saul A Teukolsky, William T Vetterling, and Brian P Flannery. *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge university press, 2007.

[322] William M Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971.

[323] Leon Danon, Albert Diaz-Guilera, and Alex Arenas. The effect of size heterogeneity on community identification in complex networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2006(11):P11010, 2006.

[324] Mark EJ Newman. Fast algorithm for detecting community structure in networks. *Physical review E*, 69(6):066133, 2004.

[325] Vesa Kuikka. Modelling community structure and temporal spreading on complex networks. *Computational Social Networks*, 8(1):13, 2021.

[326] Guiqiong Xu, Jiawen Guo, and Pingle Yang. Tns-lpa: an improved label propagation algorithm for community detection based on two-level neighbourhood similarity. *Ieee Access*, 9:23526–23536, 2020.

[327] G Bounova. Octave networks toolbox first release, 2014.

[328] Wayne W Zachary. An information flow model for conflict and fission in small groups. *Journal of anthropological research*, 33(4):452–473, 1977.

[329] David Lusseau, Karsten Schneider, Oliver J Boisseau, Patti Haase, Elisabeth Slooten, and Steve M Dawson. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology*, 54(4):396–405, 2003.

[330] Donald E Knuth and Stanford GraphBase. A platform for combinatorial computing, 1993.

[331] Valdis Krebs. Books about us politics. *unpublished, compiled by Mark Newman*, 2004.

[332] John G White, Eileen Southgate, J Nichol Thomson, and Sydney Brenner. The structure of the nervous system of the nematode caenorhabditis elegans. *Philos Trans R Soc Lond B Biol Sci*, 314(1165):1–340, 1986.

[333] Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world'networks. *nature*, 393(6684):440, 1998.

[334] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM transactions on Knowledge Discovery from Data (TKDD)*, 1(1):2–es, 2007.

[335] Hao Yin, Austin R Benson, Jure Leskovec, and David F Gleich. Local higher-order graph clustering. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 555–564, 2017.

[336] Matthew Richardson, Rakesh Agrawal, and Pedro Domingos. Trust management for the semantic web. In *The Semantic Web-ISWC 2003*, pages 351–368. Springer, 2003.

[337] TRG International. Five levels of manufacturing collaboration. `https://www.trginternational.com/resources/five-levels-of-manufacturing-collaboration`, 2019.

[338] The Aberdeen Group. Collaboration in manufacturing: Making the remote expert local, 2013.

[339] Howard H Rosenbrock. Machines with a purpose. 1990.

[340] Yuqian Lu, Juvenal Sastre Adrados, Saahil Shivneel Chand, and Lihui Wang. Humans are not machines—anthropocentric human–machine symbiosis for ultra-flexible smart manufacturing. *Engineering*, 7(6):734–737, 2021.

[341] David Romero, Ovidiu Noran, Johan Stahre, Peter Bernus, and Åsa Fast-Berglund. Towards a human-centred reference architecture for next generation balanced automation systems: human-automation symbiosis. In *IFIP International Conference on Advances in Production Management Systems*, pages 556–566. Springer, 2015.

[342] Francesco Longo, Letizia Nicoletti, and Antonio Padovano. Smart operators in industry 4.0: A human-centered approach to enhance operators' capabilities and competencies within the new smart factory context. *Computers & industrial engineering*, 113:144–159, 2017.

[343] Tamás Ruppert, Szilárd Jaskó, Tibor Holczinger, and János Abonyi. Enabling technologies for operator 4.0: A survey. *Applied sciences*, 8(9):1650, 2018.

[344] Iveta Zolotová, Peter Papcun, Erik Kajáti, Martin Miškuf, and Jozef Mocnej. Smart and cognitive solutions for operator 4.0: Laboratory h-cpps case studies. *Computers & Industrial Engineering*, 139:105471, 2020.

[345] Francesco Longo, Letizia Nicoletti, and Antonio Padovano. New perspectives and results for smart operators in industry 4.0: A human-centered approach. *Computers & Industrial Engineering*, 163:107824, 2022.

[346] Bianca Scholten. *The road to integration: A guide to applying the ISA-95 standard in manufacturing.* Isa, 2007.

[347] Nitesh Khilwani, Jennifer A Harding, and Alok K Choudhary. Semantic web in manufacturing. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 223(7):905–924, 2009.

[348] Elisa Negri, Luca Fumagalli, Marco Garetti, and Letizia Tanca. Requirements and languages for the semantic representation of manufacturing systems. *Computers in Industry*, 81:55–66, 2016.

[349] Manfredi Bruccoleri, Sergio Noto La Diega, and Giovanni Perrone. An object-oriented approach for flexible manufacturing control systems analysis and design using the unified modeling language. *International Journal of Flexible Manufacturing Systems*, 15(3):195–216, 2003.

[350] Zhenyong Wu, Jihua Liao, Wenyan Song, Hanling Mao, Zhenfeng Huang, Xinxin Li, and Hanying Mao. Semantic hyper-graph-based knowledge representation architecture for complex product development. *Computers in Industry*, 100:43–56, 2018.

[351] Song Feng, Emily Heath, Brett Jefferson, Cliff Joslyn, Henry Kvinge, Hugh D Mitchell, Brenda Praggastis, Amie J Eisfeld, Amy C Sims, Larissa B Thackray, et al. Hypergraph models of biological networks to identify genes critical to pathogenic viral response. *BMC bioinformatics*, 22(1):1–21, 2021.

[352] Cliff A Joslyn, Sinan G Aksoy, Tiffany J Callahan, Lawrence E Hunter, Brett Jefferson, Brenda Praggastis, Emilie Purvine, and Ignacio J Tripodi. Hypernetwork science: from multidimensional networks to computational topology. In *International Conference on Complex Systems*, pages 377–392. Springer, 2020.

[353] Sinan G Aksoy, Cliff Joslyn, Carlos Ortiz Marrero, Brenda Praggastis, and Emilie Purvine. Hypernetwork science via high-order hypergraph walks. *EPJ Data Science*, 9(1):16, 2020.

[354] Michael Kaufmann, Marc van Kreveld, and Bettina Speckmann. Subdivision drawings of hypergraphs. In *International Symposium on Graph Drawing*, pages 396–407. Springer, 2008.

[355] Youjia Zhou, Archit Rathore, Emilie Purvine, and Bei Wang. Topological simplifications of hypergraphs. *arXiv preprint arXiv:2104.11214*, 2021.

[356] Komal Kapoor, Dhruv Sharma, and Jaideep Srivastava. Weighted node degree centrality for hypergraphs. In *2013 IEEE 2nd Network Science Workshop (NSW)*, pages 152–155. IEEE, 2013.

[357] Cliff A Joslyn, Sinan Aksoy, Dustin Arendt, Jesun Firoz, Louis Jenkins, Brenda Praggastis, Emilie Purvine, and Marcin Zalewski. Hypergraph analytics of domain name system relationships. In *International Workshop on Algorithms and Models for the Web-Graph*, pages 1–15. Springer, 2020.

[358] Bahare Fatemi, Perouz Taslakian, David Vazquez, and David Poole. Knowledge hypergraphs: Prediction beyond binary relations. *arXiv preprint arXiv:1906.00137*, 2019.

[359] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, et al. Knowledge graphs. *Synthesis Lectures on Data, Semantics, and Knowledge*, 12(2):1–257, 2021.

[360] Xiaojun Chen, Shengbin Jia, and Yang Xiang. A review: Knowledge reasoning over knowledge graph. *Expert Systems with Applications*, 141:112948, 2020.

[361] Muhammad Akram and Anam Luqman. *Fuzzy hypergraphs and related extensions.* Springer, 2020.

[362] Musavarah Sarwar, Muhammad Akram, and Noura Omair Alshehri. A new method to decision-making with fuzzy competition hypergraphs. *Symmetry*, 10(9):404, 2018.

[363] Lászlo Nagy, Tamás Ruppert, and János Abonyi. Ontology-based analysis of manufacturing processes: Lessons learned from the case study of wire harness production. *Complexity*, 2021, 2021.

[364] László Nagy, Tamás Ruppert, and János Abonyi. Human-centered knowledge graph-based design concept for collaborative manufacturing. In *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–8, 2022.

[365] László Nagy, Tamás Ruppert, and János Abonyi. Analytic hierarchy process and multilayer network-based method for assembly line balancing. *Applied Sciences*, 10(11):3932, 2020.

[366] László Nagy, Tamás Ruppert, Andreas Löcklin, and János Abonyi. Hypergraph-based analysis and design of intelligent collaborative manufacturing space. *Journal of Manufacturing Systems*, 65:88–103, 2022.

[367] N. S. Ong and G. Boothroyd. Assembly times for electrical connections and wire harnesses. *The International Journal of Advanced Manufacturing Technology*, 6(2):155–179, 1991.

[368] NS Ong. Activity-based cost tables to support wire harness design. *International Journal of Production Economics*, 29(3):271–289, 1993.