Robotics and Computer–Integrated Manufacturing 89 (2024) 102766

Contents lists available at ScienceDirect

Robotics and Computer-Integrated Manufacturing

journal homepage: www.elsevier.com/locate/rcim

Full length article

An ontology and rule-based method for human–robot collaborative disassembly planning in smart remanufacturing

Youxi Hu [a], Chao Liu [a,*], Ming Zhang [a], Yuqian Lu [b], Yu Jia [a], Yuchun Xu [a]

[a] College of Engineering and Physical Sciences, Aston University, Birmingham, B4 7ET, UK
[b] Department of Mechanical Engineering, The University of Auckland, 0632, New Zealand

ARTICLE INFO

Keywords:
Human–Robot Collaborative Disassembly (HRCD)
Ontology
Semantic Web Rule Language (SWRL) rule
Disassembly planning
Remanufacturing

ABSTRACT

Disassembly is a decisive step in the remanufacturing process of End-of-Life (EoL) products. As an emerging semi-automatic disassembly paradigm, human–robot collaborative disassembly (HRCD) offers multiple disassembly methods to enhance flexibility and efficiency. However, HRCD increases the complexity of planning and determining the optimal disassembly sequence and scheme. Currently, the optimisation process of heuristic methods is difficult to interpret, and the results cannot be guaranteed as globally optimal. Consequently, this paper introduces a general ontology model for HRCD, along with a rule-based reasoning method, to automatically generate the optimal disassembly sequence and scheme. Firstly, the HRCD ontology model establishes the disassembly-related information for EoL products in a standardised approach. Then, customised disassembly-related rules are proposed to regulate the precedence constraints and optional disassembly methods for each disassembly task of EoL products. The optimal disassembly sequence and scheme are automatically generated by combining supportive rules with the ontology model. Lastly, the human–robot collaborative disassembly planning of a gearbox is presented as a case study to validate the feasibility of the proposed methods. Our method generates an optimal disassembly scheme compared with other heuristic algorithms, achieving the shortest process time of 308 units and the fewest number of disassembly direction change of 3 times. Additionally, the reasoning procedure can be easily tracked and modified. The proposed method is both universal and easily reproducible, allowing it to be extended to support the entire remanufacturing process.

1. Introduction

With the rapid development of Industry 4.0 technologies such as collaborative robots and Industrial Internet of Things (IIoT), a greater number of mechanical products face higher quality requirements and shorter renewal cycles [1]. During this transition period, there has been a drastic increase in the rate and volume of disposed End-of-Life (EoL) products. Traditional methods of disposal and recycling not only lead to substantial resource wastage but also aggravate environmental pollution [2]. In this context, remanufacturing has emerged as a burgeoning sector, leveraging advanced technologies and industrialised production methods to enhance the residual value and extend the life cycle of EoL products, thus achieving energy conservation and environmental protection objectives [3].

The remanufacturing process contains a series of processes, typically including cleaning, disassembly, inspection, maintenance, reprocessing, and re-assembly [4,5]. Unlike traditional repair or refurbishing methods, the performance and quality of remanufactured products must be equal to or surpass those of the original products [6]. When compared to original manufactured products, remanufactured products can achieve cost savings of up to 80%, energy savings of 60%, reduce material consumption by 70% and decrease air emissions by 85% [7]. Thus, remanufacturing of EoL products offers significant economic advantages and has become an integral component of the circular economy and the green manufacturing industry.

The disassembly process, an indispensable step in the remanufacturing process, involves separating and recycling valuable subassemblies or components from EoL products [8]. Different from the automated and intelligent assembly processes, the disassembly process is still in its early stages because of the inherent uncertainties related to the quality and failure modes of EoL products [9]. Currently, most disassembly processes are manually conducted by humans, leading to low operational efficiency and high costs. Due to the inherent uncertainties in EoL products, traditional highly-automated assembly line operations are also unsuitable for disassembly.

Integrating industrial robots offers a promising approach to enhance the automation and intelligence of the disassembly process. Equipped

* Corresponding author.
E-mail address: c.liu16@aston.ac.uk (C. Liu).

https://doi.org/10.1016/j.rcim.2024.102766
Received 5 October 2023; Received in revised form 24 January 2024; Accepted 16 March 2024
Available online 20 March 2024

Abbreviations

cBDB	canBeDisassembledBy
cBDI	canBeDisassembledIn
cBDS	canBeDisassembledSimultaneously
dC	directCover
dF	differentForm
dD	disassemblyDirection
hC	hasCost
hDA	hasDisassemblyAction
hDD	hasDisassemblyDirection
hDM	hasDisassemblyMethod
hPO	hasPartOf
hPT	hasProcessTime
hTBDA	hasToBeDisassembledAfter
hTBDDA	hasToBeDirectDisassembledAfter
iDCB	isDirectCoveredBy
iFB	isFixedBy
iPO	isPartOf

with attributes such as high precision, sensitivity, and repeatability, industrial robots are adept at handling numerous basic and repetitive disassembly tasks, thereby ensuring consistent performance [10]. However, current industrial robots cannot fully replace human involvement in the disassembly process for complete automation [11]. In situations with complex or uncertain disassembly tasks, the rigid actions of robots, which lack contextual awareness, can inadvertently damage valuable components, thus reducing their residual value [12].

In this context, human–robot collaborative disassembly (HRCD), a modern semi–automatic manufacturing paradigm, has become an ideal solution. Industrial robots can undertake basic and repetitive disassembly tasks while assisting humans in those complex disassembly tasks [13] Human labour can be equipped with detailed information about the EoL products, making timely and adaptive decisions and ensuring the smooth execution of the disassembly tasks, thereby augmenting disassembly efficiency [14]. Hence, HRCD method combines the strengths of both humans and robots, enhancing automation and intelligence while maintaining the necessary flexibility and adaptability [15]. By effectively addressing the inherent uncertainties in the disassembly process, this collaboration results in improved efficiency.

However, the HRCD planning faces several challenges:

1. Due to the diversity and uncertainty of EoL products, current product information models are limited and not suitable for HRCD. There is a lack of a standardised and universal model designed for efficiently modelling various EoL products.
2. Currently, the common methods for planning and determining the optimal disassembly sequence involve the use of heuristic optimisation algorithms. However, the execution procedures and optimal results of these algorithms are not easily interpreted and determined. It would be more reliable to propose a novel, structured, and interpretable method for planning and achieving the optimal disassembly sequence and scheme.
3. The disassembly process, inherently a divergent process, involves separating subassemblies or components from EoL products. Different subassemblies or components can be disassembled concurrently. Under the HRCD scenario, multiple disassembly methods can be applied to each task. As a result, numerous optional disassembly methods and sequences exist, making it highly challenging to determine the optimal disassembly scheme, especially for complex products.

To address these challenges, this study proposes a general ontology model and rule-based method for HRCD planning of EoL products. This disassembly-related ontology model is used to store disassembly-related knowledge for every component within EoL products, providing a standard and structured knowledge representation model for human–robot collaborative disassembly. Disassembly knowledge can be structurally organised and managed in a standardised manner, and quickly extracted from extensive data resources associated with EoL products. On this basis, a Semantic Web Rule Language (SWRL) rule-based human–robot collaborative disassembly knowledge reasoning method has been developed to reason out the precedence constraints and optional disassembly methods for each disassembly task. Subsequently, this method generates the optimal disassembly scheme. Lastly, the human–robot collaborative disassembly planning of a gearbox is presented as a case study to validate the feasibility of the proposed methods. The overall workflow of this research is presented in Fig. 1.

The rest of this paper is organised as follows: Section 2 reviews relevant literature. Section 3 presents the establishment of the human–robot collaborative disassembly ontology and introduces the semantic model for EoL products. Section 4 describes the proposed rules for reasoning and generating the optimal disassembly scheme. Section 5 presents a case study for verifying the feasibility and effectiveness of the proposed method. Section 6 offers discussions, while Section 7 provides conclusions and outlines of future work.

2. Literature review

The literature review is summarised from three aspects: human–robot collaborative disassembly, ontology-based product information models, and rule-based reasoning for disassembly planning. These three aspects cover the background and methodologies related to this study. At the end of this section, the research gaps and challenges associated with implementing human–robot collaborative disassembly are discussed.

2.1. Human–robot collaborative disassembly

In line with definitions from intelligent manufacturing, human–robot collaboration in smart remanufacturing is defined as an interactive environment where humans and industrial robots coexist, sharing the same workspace, resources, and remanufacturing tasks. While humans primarily control and monitor the remanufacturing processes [16], industrial robots, endowed with environmental sensing, cognitive capabilities, and relevant knowledge, are positioned to closely assist humans in accomplishing the remanufacturing tasks, or operate autonomously. The advantage of human–robot collaboration lies in the ability of these robots to handle high-load, repetitive, and hazardous tasks while ensuring human safety [17]. This not only improves overall production efficiency but also substantially reduces the workload and stress on humans. Compared to the traditional human disassembly process, human–robot collaboration in disassembly can improve overall efficiency by combining the automation and intelligence with the human's knowledge and expertise [18,19].

Liu et al. [20] integrated advanced technologies such as Cyber–Physical Production Systems (CPPS) and Artificial Intelligence (AI) to establish a comprehensive human–robot collaborative disassembly system framework. They validated the feasibility and efficiency of the system through case studies involving human–robot collaborative disassembly task planning, distance-based safety strategies, and motion-driven control methods. Huang et al. [21] introduced an active compliance control method for the human–robot collaborative disassembly of press-fit components. They demonstrated the feasibility of their approach with a case study in which a human and robot collaboratively disassembled an automotive water pump. In their follow-up study, they designed a human–robot collaboration paradigm comprising two collaborative robots and an operator, and validated it using

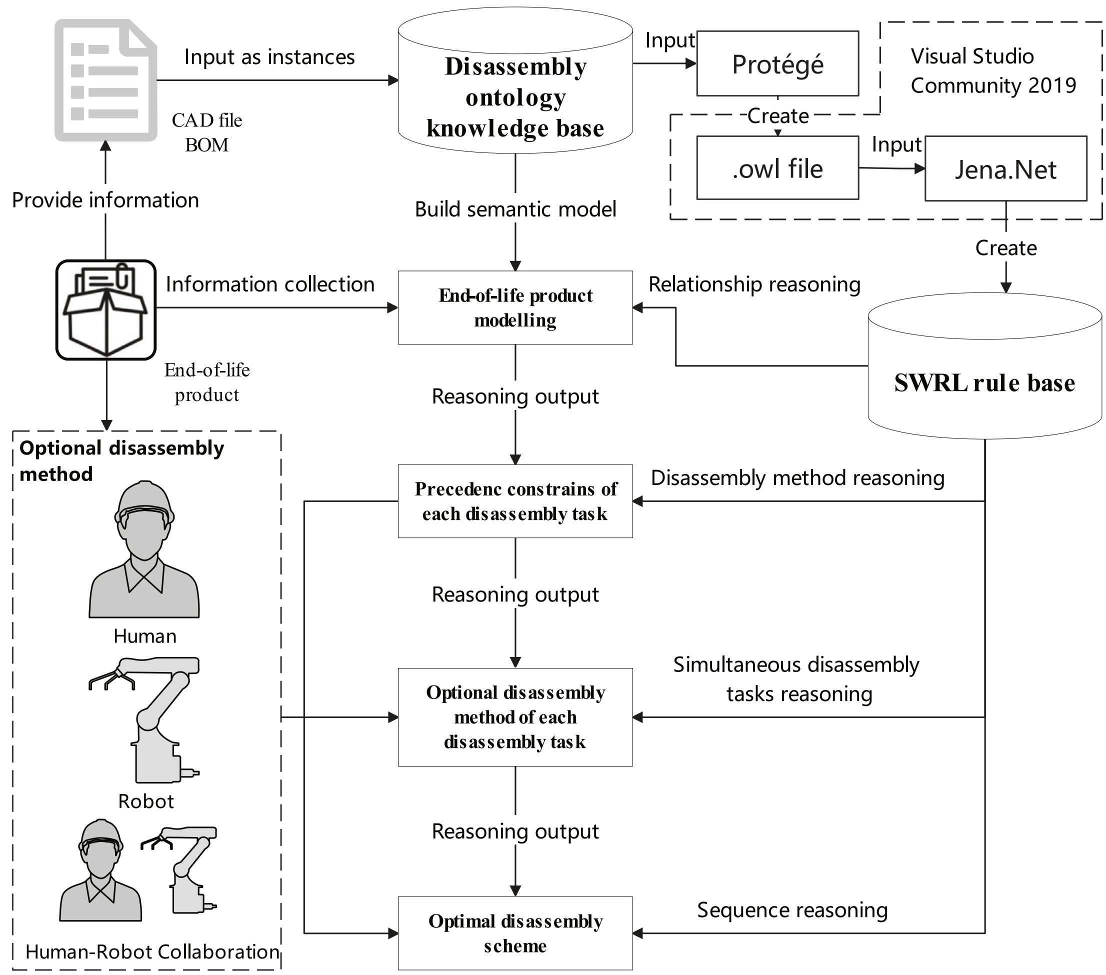


Fig. 1. Overall workflow of this research.

the same method [22]. Lee et al. [23] proposed a disassembly sequence planning algorithm for the human–robot collaborative disassembly environment. Considering constraints such as limited resources and worker safety, the proposed algorithm aims to reduce the overall disassembly time. The effectiveness of the proposed method was validated through a case study involving the disassembly of a disposed hard disk drive. Xu et al. [24] introduced the Pareto-based modified discrete bees algorithm (MDBA-Pareto) to address the disassembly sequence planning problem in human–robot collaborative settings. This method considers multiple optimisation objectives, including disassembly time, cost, and difficulty. By employing computer disassembly as a case study and comparing their method with other relevant algorithms, they demonstrated the effectiveness of their proposed approach. Parsa and Saadat [25] classified human–robot collaboration tasks by evaluating the remanufacturing capability of EoL product components. This enriched the definitions of collaboration categories within human–robot collaboration. Subsequently, they generated a near-optimal disassembly sequence using an enhanced genetic algorithm. The efficiency of their approach was validated by comparing it to the particle swarm optimisation algorithm. Elguea-Aguinaco et al. [26] introduced a goal-conditioned reinforcement learning approach to ensure real-time collision avoidance, facilitating safe interactions in the human–robot disassembly process. Chu and Chen [27] proposed a hybrid particle swarm optimisation algorithm based on Q-learning to address human–robot collaborative disassembly challenges of power batteries. By comparing their proposed algorithm with other related meta-heuristic approaches, they affirmed its effectiveness. Guo et al. [28] developed a method for human–robot collaborative partial-destructive disassembly sequence planning, considering multiple failure modes of EoL products. They employed a multi-layer chromosome encoding technique with the aim of determining the optimal disassembly sequence.

The majority of papers addressing the HRCD issue focus on developing optimisation algorithms for disassembly sequence planning, aiming to identify the optimal disassembly sequence [29]. However, given that disassembly sequence planning is inherently an NP-hard problem, it becomes theoretically impossible to determine the optimal disassembly sequence through those optimisation algorithms [30]. Consequently, there is a need to consider incorporating alternative methods, such as graph theory, and knowledge reasoning, to plan and determine the optimal human–robot collaborative disassembly sequences.

2.2. Ontology-based product information model

The product information model, which represents products and associated disassembly data in a structured format, serves as the premise for disassembly sequence planning [31]. For effective disassembly sequence planning, it is essential to construct a well-defined, comprehensive product information model for EoL products. This model should be able to offer a shared, scalable, and organised information structure in a designated format [32]. Two primary models are predominantly used at the current stage: matrix-based models and graph-based models. Both models can properly represent the connection relationships and precedence constraints among components in EoL products [33], intuitively generating disassembly sequences. However, these two models are not suitable for storing and transferring other disassembly-related knowledge, such as the required direction, action, or tool for each component's disassembly. This drawback significantly hampers the expansion of disassembly knowledge and reduces the quality of disassembly planning solutions. To compensate for this limitation, the product information model requires a more standardised, structured, and intelligent method to build upon.

Knowledge engineering has been widely employed for knowledge acquisition and sharing in manufacturing [34]. It stores and shares knowledge in the form of ontologies. Ontologies, serving as tools for building conceptual models and expressing semantic knowledge, have been widely deployed in fields like artificial intelligence and systems engineering [35]. The ontology-based product information model is capable of representing knowledge in a more standardised and structured manner [36]. It facilitates the easy storage and access of various disassembly-related knowledge, including product hierarchical structures, connection constraints, disassembly rules, and selection criteria for disassembly directions, actions, tools, etc.

Over the past few years, ontology-based models for assembly have garnered significant attention for their potential in enhancing the assembly process's efficiency and intelligence. According to Qiao et al. [37] and Zhong et al. [38], ontologies can capture complex interrelationships among components and assembly processes, thereby facilitating more effective and automated assembly sequence planning. This sentiment is further demonstrated by Gong et al. [39], who emphasised the importance of semantic representations in reducing assembly errors and reusing both process knowledge and assembly sequence planning experience.

Moreover, it possesses good scalability, allowing timely adjustments to meet different scenarios in disassembly. Zhu and Roy [40,41] developed a disassembly information model that includes various types of knowledge related to EoL products, such as product hierarchical structure, feasible disassembly sequences, component uncertainties, and degradation information. Building on this, they aimed to generate more reasonable disassembly sequences. Foo et al. [42,43] proposed an ontology-based structural model to manage the disassembly-related knowledge of EoL products. They employed an artificial learning method for component recognition during disassembly and validated its efficacy using the disassembly of LCD monitors as a case study.

The consensus in the literature indicates a promising future for ontology-based assembly models, as they pave the way for more intelligent, adaptable, and efficient manufacturing processes. However, existing disassembly information models have not been designed for human–robot collaborative disassembly scenarios. Moreover, these models are relatively simplistic and unsuitable for the disassembly process in remanufacturing. Therefore, there is a need to develop a more comprehensive ontology-based model for EoL products, specifically tailored to human–robot collaborative disassembly in remanufacturing.

2.3. Rule-based reasoning for disassembly sequence planning

Rule-based reasoning inherently operates by constructing pertinent semantic rules or processing mechanisms to extract tacit knowledge hidden within explicit knowledge [44]. Furthermore, knowledge reasoning can resolve inconsistencies within the product information model and detect contradictions present within the existing knowledge [45].

Veerakamolmal and Gupta [46] proposed a case-based reasoning method to automatically plan and generate the disassembly sequence. Giudice [47] proposed a rule-based approach to reason the difficulty of spatial and junction constraints of components. Consequently, this approach supports determining the optimal disassembly depth and enhances the disassemble ability of EoL products. Chen et al. [48] proposed a system based on ontology and case-based reasoning method to realise the automatic disassembly decision-making and reduce costs. Yu et al. [49] developed an ontology and partial destructive rule-based method, which automated planning the disassembly sequence of disposed automotive traction batteries (ATB).

The rule-based reasoning method offers a range of distinct advantages, especially its structured approach to problem-solving. Tor et al. [50] proposed a rule-based representation approach for the functional design of mechanical products. Its deterministic nature ensures that, given a specific input, the output remains consistent, thereby reducing uncertainty in decision-making processes related to physical behaviours. Such consistency leads to more straightforward debugging and validation of processes. Zheng et al. [51] introduced a knowledge-based engineering method for designing the architectures of robotic manufacturing systems. Within this method, a rule-based reasoning process is outlined to describe the explicit semantic information of the components of robotic manufacturing systems [52]. Integrating expert knowledge in the form of predefined rules guarantees that the system operates based on tried and tested expertise, laying a foundation for reliability. Additionally, Reddy and Fields [53] highlighted two other advantages of the rule-based reasoning method through his review paper:

1. The transparency of rule-based systems means that decisions can be traced back to specific rules, offering enhanced interpretability and understandability. This feature is especially vital in complex systems where grasping the logic behind decisions is essential.
2. Rule-based reasoning can be effortlessly expanded by adding new rules without necessarily modifying existing ones, which supports scalability and adaptability.

In summary, the rule-based reasoning method presents a clear, scalable, and reliable approach to automated reasoning and decision-making. It is evident that rule-based reasoning for disassembly planning primarily relies on pre-established product information semantic models. Therefore, the existing rules are non-transferable and unsuitable for contexts involving human–robot collaborative disassembly.

2.4. Brief summary

The existing ontology model has two limitations, making it unsuitable for human–robot collaborative disassembly in remanufacturing:

1. Current disassembly ontology models are relatively simple, suggesting only two types of components in EoL products [48,49]. This limited scope fails to effectively differentiate among various components, thereby impeding the decision-making process in disassembly sequence planning.
2. Moreover, current disassembly ontology models do not consider or establish a human–robot collaborative working environment and lack a related knowledge base for robots.

Based on the human–robot collaborative disassembly ontology model, there is also a need to formulate corresponding rules to reason and generate the optimal disassembly sequence scheme. In this context, this research constructs a human–robot collaborative disassembly ontology model and determines the optimal disassembly sequence scheme based on the ontology and SWRL rules. The major contributions of this paper are as follows:

1. Proposed a general ontology model that supports rule-based reasoning for human–robot collaborative disassembly in remanufacturing. The model introduced in this research proposes three distinct categories of components for EoL products, making it easier to segregate and identify the value of each component. Consequently, this aids in the disassembly sequence planning of EoL products.
2. Developed a rule-based method for planning and generating human–robot collaborative disassembly sequences. Feasible disassembly sequences can be effectively generated through an iterative reasoning process, avoiding the generation of a large number of useless and redundant solutions.

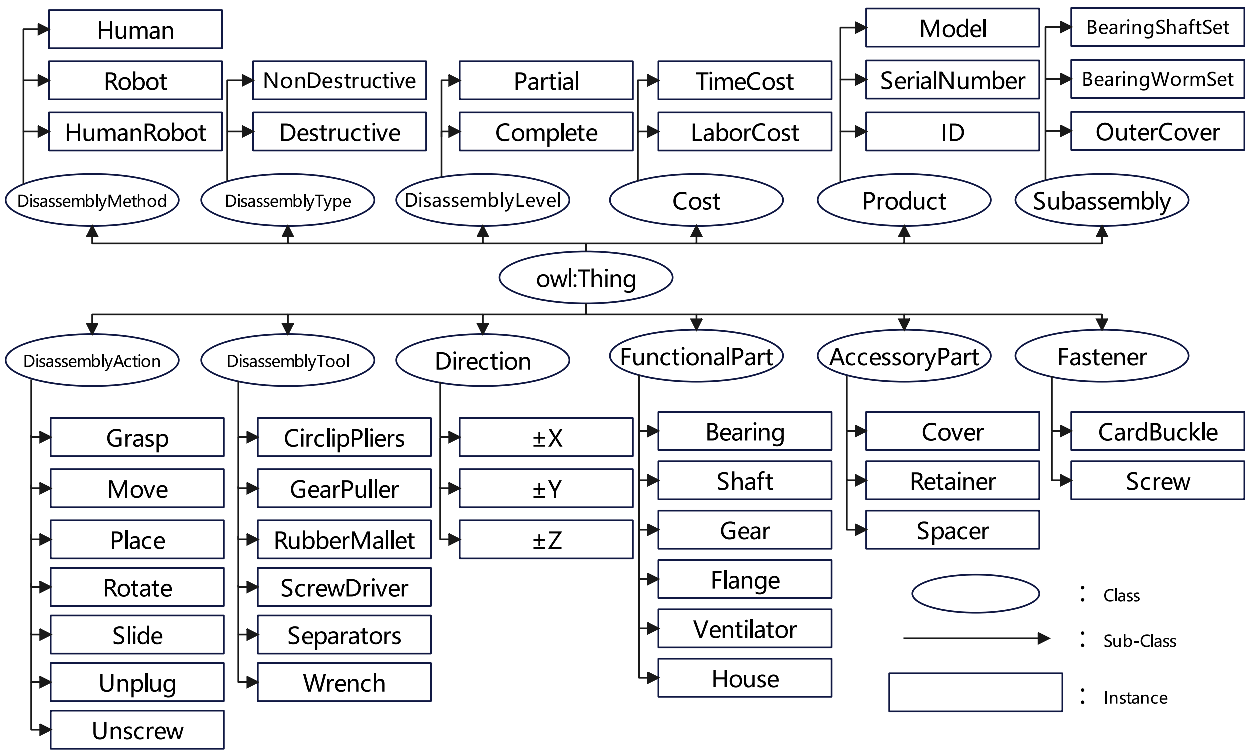


Fig. 2. The class hierarchy of the proposed human–robot collaborative disassembly ontology.

3. Beyond considering precedence constraints from the disassembly ontology model, the method also integrates disassembly-related knowledge, including disassembly directions, tools, and actions. Based on this, the method selects components that can be executed concurrently through rule-based reasoning, subsequently determining the optimal disassembly sequence scheme.

3. Disassembly ontology and product semantic model

In this section, the human–robot collaborative disassembly ontology is introduced. It formalises and semantically represents the disassembly-related information within the EoL product.

3.1. Disassembly ontology

A domain ontology is designed to describe the knowledge of a specific domain. It provides a semantic description of concepts within the specific domain and the relationships among those concepts [54]. The goal of constructing a domain ontology is to capture related knowledge, thereby helping eliminate ambiguity and redundancy in both concepts and terminologies. A specific domain ontology establishes a standard, shared, and unified understanding of specific domain knowledge [55]. The construction of an ontology can be broken down into definitions of a series of classes, object properties, and data properties [56]. A class is defined as a group of instances that share the same properties or characteristics. Each class can be interconnected with and interact with other classes through object properties. Instances, which serve as the fundamental elements of the ontology, are the objects [57].

The class hierarchy of the proposed human–robot collaborative disassembly ontology is shown in Fig. 2. The top-level class that represents all possible things or entities is termed as *owl:Thing*. There are twelve subclasses of *owl:Thing* proposed for covering the knowledge for human–robot collaborative disassembly, offering a controlled vocabulary and structure for the upper ontology based on fundamental concepts.

Two subclasses are proposed to describe the EoL product-level structure:

1. *Product* (*p*) represents an abstract concept of an EoL product set, serving as the research target in remanufacturing. It can refer to any potential EoL mechanical product.
2. *Subassembly* (*sa*) represents a category of components assembled from combining various components that can function as a unit and is a part of a larger assembly or a product. A *sa* is often constructed in a specific sequence during the manufacturing process. During the disassembly process, a *sa* is treated as a single component within the EoL product, thus reducing the number of components and optional disassembly sequences of EoL products.

The two subclasses mentioned above are frequently employed in product-level modelling. Differing from existing classifications and modelling methods at the component level, this study introduces three distinct categories for modelling the components of EoL products, to meet the performance requirements of the disassembly process in remanufacturing.

1. *FunctionalPart* (*fp*) is a category of core components that perform the main functions and are essential for the operation and performance of the product. These components possess high residual and remanufacturing value. Typically, in the remanufacturing process, the *fp* is the primary target to be reclaimed, reprocessed, and reused in remanufactured products.
2. *AccessoryPart* (*ap*) is a category of supplementary components with specific functions that can be added to a product to enhance its features, convenience, appearance, or safety. However, *aps* are not essential to the core function or performance of the product. *aps* have characteristics of good repairability and interchangeability. Typically, in the remanufacturing process, *aps* are reprocessed through additive manufacturing to upgrade the grades and qualities of the remanufactured products.
3. *Fastener* (f) is a category of hardware devices that mechanically joins or affixes two or more components together. fs are used in a wide range of applications to hold, connect or fix *fps* or *aps*

together in a secure but non-permanent manner. This allows for *fp*s or *ap*s to be disassembled without a destructive process.

The mentioned subclasses are employed to hierarchically describe the connection structure and relationship of components in EoL products. Following subclasses are proposed to manage disassembly-related knowledge within a human–robot collaborative context. The other seven subclasses proposed in this ontology model are:

1. *DisassemblyAction* (*DA*) includes various actions required for component removal, such as grasping, moving, and placing. Both humans and robots can perform these disassembly actions, and all disassembly tasks can be executed through a single action or a combination of actions.
2. *DisassemblyTool* (*DTl*) represents the tools used for separating the connections of components in EoL products, such as screwdrivers, separators, and pullers. Similarity, both humans and robots have the ability to use these disassembly tools, and each type of disassembly task typically requires specific disassembly tools.
3. *DisassemblyMethod* (*DM*) refers to three different approaches adopted for executing disassembly tasks: human, robot, and human–robot collaborative. The optional *DM*s for each disassembly task are inferred by the required *DA*s and *DM*s.
4. *DisassemblyType* (*DTy*) includes non-destructive and destructive disassembly. While destructive disassembly can cause irreversible damage and diminish the residual value of components, non-destructive disassembly is preferred in remanufacturing.
5. *DisassemblyLevel* (*DL*) indicates the depth of disassembly process, which mainly includes complete and partial disassembly. The target components within EoL products determine the disassembly level, thereby influencing the planning of the disassembly sequence.
6. *Direction* (*Dir*) represents the movement constraints of components on six coordinate axis directions ($\pm X, \pm Y, \pm Z$).
7. *Cost* (*C*) contains the time and labour expenses associated with the execution of the disassembly task through various *DM*s, each applying different *DA*s and utilising different *DTl*s.

3.2. Object and data properties of disassembly ontology

The Web Ontology Language (OWL) is a recommended language for representing ontologies in the context of semantic web standards [58]. It extends the Resource Description Framework (RDF) Schema to better represent intricate classes, attribute characteristics, and property constraints. The OWL offers rich semantic descriptions and logical reasoning capabilities, reducing redundancy in knowledge representation and promoting knowledge sharing and semantic operations. These capabilities are particularly valuable for representing complex knowledge in the domain of human–robot collaborative disassembly in remanufacturing [59]. Additionally, ontologies expressed in OWL are machine-readable and computationally friendly, making them suitable for storage and development. In the OWL-based framework, object properties delineate the attributes and inter-relational constraints of classes. The class relationships of the proposed human–robot collaborative disassembly ontology are presented in Fig. 3.

The inner circle of this figure illustrates the semantic expression of the structural relationships at the component level of the EoL product. Essentially, the structural interrelationships among all three categories of components within the EoL product can be characterised by '*isDirectCoveredBy*' (*iDCB*) and '*isFixedBy*' (*iFB*). The *iDCB* relationship indicates that a *f*, an *ap*, or a *fp* is directly covered by another *ap* or *fp*. The *iFB* relationship represents that *ap*s or *fp*s are fastened by a *f*.

To illustrate the product level more effectively and clearly, all categories of components are collectively referred to as *component* (*cp*) in this figure. Relationships within a *p* that include *sa* are denoted by *hasSubassembly* (*hsa*), while *cp* contained within a *p* or *sa* are indicated by *hasComponent* (*hc*). It is important to note that one *sa* can also function as a component for another *sa*.

Table 1
Object properties in the human–robot collaborative disassembly ontology.

No.	Object property	Domain	Range	Inverse property
1	dC_plusX	*fp, ap, sa*	*f, fp, ap, sa*	iDCB_minusX
2	dC_plusY	*fp, ap, sa*	*f, fp, ap, sa*	iDCB_minusY
3	dC_plusZ	*fp, ap, sa*	*f, fp, ap, sa*	iDCB_minusZ
4	iDCB_plusX	*f, fp, ap, sa*	*fp, ap, sa*	dC_minusX
5	iDCB_plusY	*f, fp, ap, sa*	*fp, ap, sa*	dC_minusY
6	iDCB_plusZ	*f, fp, ap, sa*	*fp, ap, sa*	dC_minusZ
7	fix_plusX	*f*	*fp, ap, sa*	iFB_minusX
8	fix_plusY	*f*	*fp, ap, sa*	iFB_minusY
9	fix_plusZ	*f*	*fp, ap, sa*	iFB_minusZ
10	iFB_plusX	*fp, ap, sa*	*f*	fix_minusX
11	iFB_plusY	*fp, ap, sa*	*f*	fix_minusY
12	iFB_plusZ	*fp, ap, sa*	*f*	fix_minusZ
13	iPO	*f, fp, ap, sa*	*sa, p*	hPO
14	cBDI	*f, fp, ap, sa*	*Dir*	N/A
15	hDA	*f, fp, ap, sa*	*DA*	N/A
16	cBDB	*f, fp, ap, sa*	*DTl*	N/A
17	hDM	*f, fp, ap, sa*	*DM*	N/A
18	hTBDA	*f, fp, ap, sa*	*f, fp, ap, sa*	N/A
19	hTBDDA	*f, fp, ap, sa*	*f, fp, ap, sa*	N/A
20	hPT	*f, fp, ap, sa*	*C*	N/A
21	hC	*f, fp, ap, sa*	*C*	N/A
22	cBDS	*f, fp, ap, sa*	*f, fp, ap, sa*	N/A
23	dF	*f, fp, ap, sa*	*Dir, DTl*	N/A

The additional disassembly-related classes are also interconnected and applied through semantic expressions. The six movement *Dir* constraints of *sa* and *cp* are characterised by *isConstrainedIn* (*iCI*). The *DL* and *DTy* of a *p* can be described through *hasDisassemblyLevel* (*hDL*) and *hasDisassemblyType* (*hDTy*), respectively. Within this framework, *sa* is a subclass of *DL*. When *sa*=ø, the *DL* is regarded as a complete disassembly. The *cp* is defined by its required *DTl*, *DA*, *DM*, and *cp* through the object properties *hasDisassemblyTool* (*hDTl*), *hasDisassemblyAction* (*hDA*), *hasDisassemblyMethod* (*hDM*), and *hasCost* (*hC*), respectively.

Moreover, in this research, the *DM* primarily encompasses three different approaches: human, robot, and human–robot collaborative. The selection of an optional *DM* is determined by the required *DTl* and *DA* based on the disassembly task for each *cp*. The *DM* and *C* for a *cp* are dependent on the *DTl* and *DA* through the following rule-based reasoning method. Building on the ontology object properties presented in articles [48,49], this study proposes the object properties for human–robot collaborative disassembly ontology, as shown in Table 1.

In this table, the *domain* of an object property specifies the class of individuals to which the property can be applied and the *range* of an object property defines the class of individuals that can be the value of the property. The *domain* and *range* are used to describe the types of things that properties can relate to in an ontology, ensuring that the data adheres to the logical structure defined by the ontology. The proposed object properties are:

- Object properties 1–12 characterise the assembly relationships of product components along the six coordinate axis directions (+X/−X/+Y/−Y/+Z/−Z). These properties collectively determine the full assembly structure of the product. Specifically, object properties 1–6 describe the coverage relationships among different components, while object properties 7–12 determine the fastening relationships between fasteners and both functional and accessory parts.
- Object property 13 represents the hierarchical relationship among products, subassemblies, and individual components.
- Object property 14 indicates the potential disassembly directions of a subassembly or component in an EoL product.

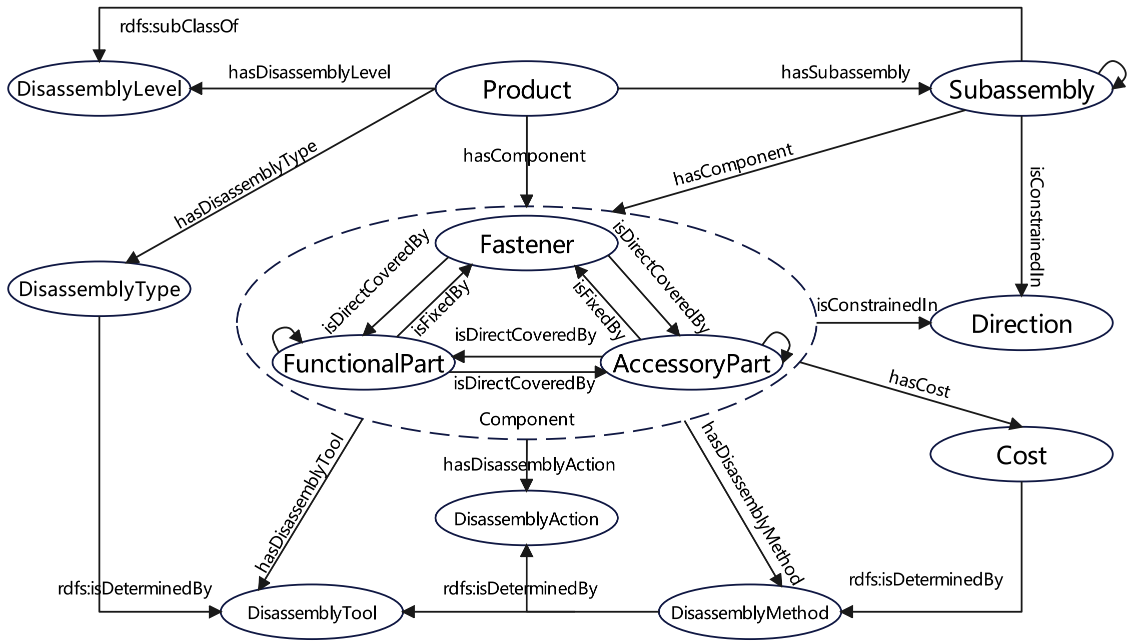


Fig. 3. The class relationships of the proposed human–robot collaborative disassembly ontology.

Table 2
Data properties in the human–robot collaborative disassembly ontology.

No.	Data property	Domain	Range	Description
1	cBDI_plusX cBDI_plusY cBDI_plusZ cBDI_minusX cBDI_minusY cBDI_minusZ	*f, fp, ap, sa*	boolean	Record the optional disassembly direction of a f, fp, ap or sa
2	iD	*f, fp, ap, sa, p*	string	Record the identity of a *f, fp, ap, sa* or *p*
3	nameOfComponent	*f, fp, ap, sa*	string	Record the name of a *f, fp, ap* or *sa*
4	quantity	*f, fp, ap*	int	Record the quantity of a *f, fp, ap* or *sa* in the *p*
5	humanProcessTime robotProcessTime humanRobotProcessTim	*f, fp, ap, sa*	float	Record the process time of disassemble a *f, fp, ap or sa* using human, robot or human–robot *DM*

- Object properties 15–17 correspondingly denote the required disassembly actions, tools, and feasible disassembly methods for each disassembly task.
- Object properties 18–19 elucidate the disassembly precedence relationships between subassemblies and components, outlining the disassembly sequence for the product. This property exhibits transitivity. Components with direct disassembly precedence can be considered as subassemblies.
- Object properties 20–21 convey the time and operational costs required for the disassembly of subassemblies and components.
- Object property 22 indicates subassemblies or components that can be disassembled concurrently.
- Object property 23 is primarily used to assess whether different subassemblies or components can be disassembled simultaneously. *Dir* and *DTl* serve as two evaluation indices, corresponding to Auxiliary rules 67–69 presented in Table 5.

Data properties are employed to characterise the features, attributes, or other data-related information of entities within various classes [59]. Analogous to object properties, each data property has its corresponding domain and range. The definitions and descriptions of pertinent data properties are presented in Table 2.

3.3. An illustrative example

In the evolving landscape of smart remanufacturing, the disassembly of EoL products emerges as a critical facet, demanding systematic and structured approaches. The product semantic model for disassembly emerges as a crucial innovation, acting as a linchpin to bridge the gap between EoL products and the subsequent remanufacturing processes. This model focuses on a comprehensive representation of products, encapsulating not only their physical structural relationship and attributes but also the associated information such as maintenance log, failure mode, etc. By leveraging semantic technologies, this model captures intricate relationships, dependencies, and hierarchies among EoL product components. By charting the hierarchical and connection-based relationships within product structures, the semantic model facilitates a more informed, streamlined, and efficient disassembly process.

The product semantic model is part of the proposed human–robot collaborative disassembly ontology model, which focuses on describing the components and their interrelationships in EoL products. In this section, we primarily aim to illustrate the product semantic model through the example of a belt roller support assembly. Fig. 4 presents

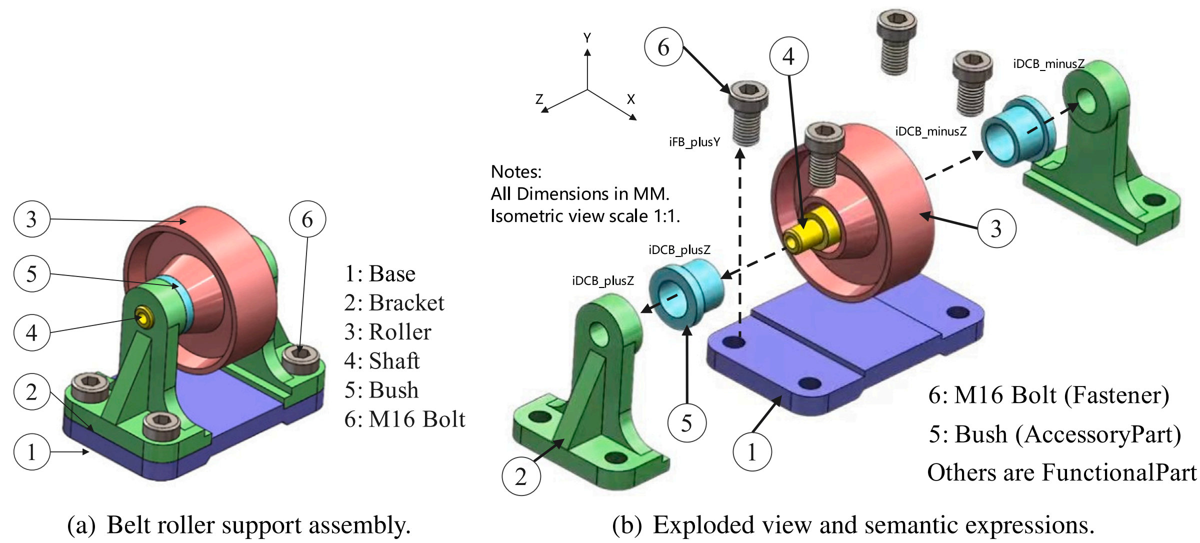


Fig. 4. An illustrative example.

the relationship between the proposed three categories of the components in the belt roller support assembly: bolts (in grey) are fasteners, bushes (in light blue) are accessory parts, while the base (in navy blue), shaft (in yellow), bracket (in green) and roller (in red) are functional parts. Based on the product semantic model and the object properties in Tables 1, 2 and Fig. 3, we have established the topological structures of the belt roller support assembly, which incorporate connections and hierarchical relationships. The following are semantic expressions and descriptions of the belt roller support assembly:

- Bracket iFB_plusY bolt: brackets are fixed by bolts from +Y direction.
- Base iDCB_plusZ bracket: the base is directly covered by brackets from +Y direction.
- Bush iDCB_plusZ bracket: the left bush is directly covered by the left bracket from +Z direction.
- Bush iDCB_minusZ bracket: the right bush is directly covered by the right bracket from +Z direction.
- Shaft iDCB_plusZ bracket: the shaft is directly covered by the left bracket from +Z direction.
- Shaft iDCB_minusZ bracket: the shaft is directly covered by the right bracket from +Z direction.
- Roller iDCB_plusZ shaft: the roller is directly covered by the shaft from +Z direction.

Normally, the −Y direction, which is associated with gravity, is not considered in the disassembly process.

4. Rule-based reasoning for disassembly sequence planning

In this section, the Semantic Web Rule Language (SWRL) and Semantic Query-enhanced Web Rule Language (SQWRL) are adopted for proposing the rule-based reasoning method. The first subsection presents rules related to product structure, which reason and generate precedence constraints for disassembling subassemblies or components of an EoL product. The second subsection presents rules to determine the disassembly method of each component and generate an optimal human–robot collaborative disassembly scheme.

4.1. Rules for reasoning disassembly precedence constraints

Similar to the established disassembly relationship between *fp* and *f* in [48,49], this research takes into account the disassembly relationships among *fp*, *ap*, and *f*. Disassembly tasks are executed from five coordinate axes directions (except −Y direction). The rules and descriptions for generating disassembly precedence constraints are shown in Table 3. Typically, in mechanical products, there is no situation where *f* s are fastened to one another, thus, it is not considered. There are three relationships defined as direct disassembly: *f* /*fp*, *f* /*ap*, and *ap*/*ap*. These relationships ensure consistent disassembly habits and actions and also reduce the number of changes in disassembly direction. Based on the formulated rules, it is possible to realise the precedence constraints among *fp*, *ap*, and *f* within an EoL product.

The belt roller is used as a case analysis to validate the correctness and completeness of the disassembly precedence constraint set. Utilising the CAD file (available through the provided open-source link in Appendix B) and the relationship semantic descriptions of the belt roller in Section 3.3, the actual disassembly procedures and the precedence constraint graph are presented in Fig. 5. The disassembly process involves first removing the bolts, followed by the brackets, bushes, and shaft. During this disassembly process, the roller and base separate automatically. There are 9 precedence constraints among the belt roller components, as depicted in the figure. While multiple disassembly sequences are possible, the precedence constraint set is unique and fixed, and must be adhered to throughout the disassembly process.

The product ontology and instances of the belt roller have been developed using Protégé 5.5.0, with HermiT as the embedded reasoner. There are 9 assertions generated through these rules:

- ‘Bracket_left hTBDDA Bolts’: The left bracket, directly fixed by bolts from the +Y direction, has to be direct disassembled after removing the bolts (from rules 25–30)
- ‘Bracket_right hTBDDA Bolts’: The right bracket, directly fixed by bolts from the +Y direction, has to be direct disassembled after removing the bolts (from rules 25–30).
- ‘Bush_left hTBDA Bracket_left’: The left bush, directly covered by the left bracket from the +Z direction, has to be disassembled after the left bracket (from rules 13–18).
- ‘Bush_right hTBDA Bracket_right’: The right bush, directly covered by the right bracket from the −Z direction, has to be disassembled after the right bracket. (from rules 13–18).
- ‘Shaft hTBDA Bush_left’: The shaft, directly covered by the left bush from the +Z direction, has to be disassembled after the left bush. (from rules 19–24).
- ‘Shaft hTBDA Bush_right’: The shaft, directly covered by the right bush from the −Z direction, has to be disassembled after the right bush. (from rules 19–24).

Table 3
Rules for generating the precedence constrains of components.

No.	SWRL/SQWRL	Description
1–6	*f* (?f) ˆ*fp* (?fp) ˆiDCB_*Dir* (?f, ?fp) → hTBDDA (?f, ?fp)	If a *f* is directly covered by a *fp* in any disassembly-direction of the *f*, then the *f* shall be direct disassembled after the *fp*
7–12	*f* (?f) ˆ*ap* (?ap) ˆiDCB_*Dir* (?f, ?ap) → hTBDDA (?f, ?ap)	If a *f* is directly covered by an *ap* in any disassembly-direction of the *f*, then the *f* shall be direct disassembled after the *ap*
13–18	*ap* (?ap) ˆ*fp* (?fp) ˆiDCB_*Dir* (?ap, ?fp) → hTBDA (?ap, ?fp)	If an *ap* is directly covered by a *fp* in any disassembly-direction of the *ap*, then the *ap* shall be disassembled after the *fp*
19–24	*fp* (?fp) ˆ*ap* (?ap) ˆiDCB_*Dir* (?fp ?ap) → hTBDA (?fp, ?ap)	If a *fp* is directly covered by an *ap* in any disassembly-direction of the *fp*, then the *fp* shall be disassembled after the *ap*
25–30	*fp* (?fp) ˆ*f* (?f) ˆiFB_*Dir* (?fp, ?f) → hTBDDA (?fp, ?f)	If a *fp* is fixed by a *f* in any direction, then the *fp* shall be direct disassembled after the *f*
31–36	*ap* (?ap) ˆ*f* (?f) ˆiFB_*Dir* (?ap, ?f) → hTBDDA (?ap, ?f)	If an *ap* is fixed by a *f* in any direction, then the *ap* shall be direct disassembled after the *f*
37–42	*fp* (?fp) ˆ*fp* (?fp1) ˆiDCB_*Dir* (?fp, ?fp1) → hTBDA (?fP, ?fp1)	If a *fp* A is directly covered by a *fp* B in any direction, then the *fp* A shall be disassembled after the *fp* B
43–48	*ap* (?ap) ˆ*ap* (?ap1) ˆiDCB_*Dir* (?ap, ?ap1) → hTBDDA (?aP, ?ap1)	If an *ap* A is directly covered by an *ap* B in any direction, then the *ap* A shall be disassembled after the *ap* B

Dir represents *plusX, minusX, plusY, plusZ, minusZ*, respectively.

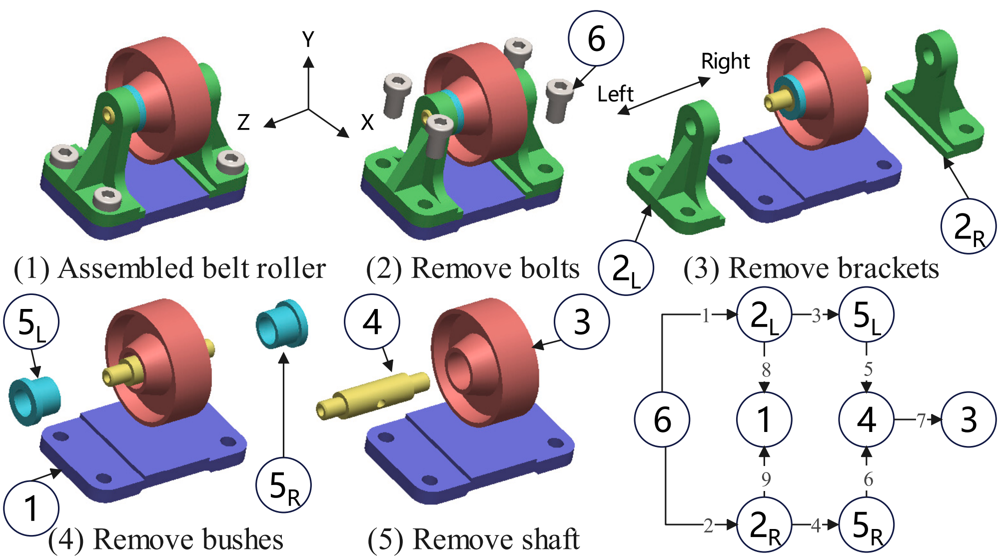


Fig. 5. Disassembly procedures and precedence constraints of the belt roller.

- 'Roller hTBDA Shaft': The roller, directly covered by shaft from the +Z direction, has to be disassembled after the shaft. (from rules 37–42).
- 'Base hTBDA Bracket_left': The base, directly covered by the left bracket from the +Y direction, has to be disassembled after the left bracket. (from rules 37–42).
- 'Base hTBDA Bracket_right': The base, directly covered by the right bracket from the +Y direction, has to be disassembled after the right bracket. (from rules 37–42).

In this case analysis, one assertion is generated from the chosen rule in accordance with its corresponding direction. The correctness and completeness of the precedence constraint set can be analysed from:

1. Correctness of the precedence constraint set: The proposed SWRL rules avoid contradictions, and all reasoning outcomes derived from these rules are valid within the context of precedence constraints. The generated precedence constraints for the belt roller accurately infer the actual component precedence constraints.
2. Completeness of the precedence constraint set: This set, a combination of outcomes from the proposed SWRL-based rules, includes 9 precedence constraints. This number matches the actual precedence constraint set of the belt roller.

The disassembly precedence constraint set generated through the proposed SWRL rules 1–48 aligns with the actual precedence constraint set of the belt roller. Although, rules 1–12, 31–36, and 43–48 do not establish any precedence constraints among the components, due to the absence of certain component relationships and semantic descriptions in the belt roller. These rules are not redundant, as they are proposed to describe and cover all possible cases or inferences, including existing potential relationships within an EoL product.

4.2. Rules for determining disassembly method

Collaborative robots are capable of managing repetitive and simple tasks efficiently in human–robot collaboration environments, which compensates for the deficiencies of manual disassembly, such as lower efficiency and higher costs. However, their effectiveness is affected by several factors, including worker safety, the complexity of the disassembly tasks, and the limitations of tools and resources. For complex disassembly tasks, humans are noted to be more flexible and efficient than robots. This is attributed to human abilities such as detection, observation, thinking, and handling, which robots currently lack. Therefore, complex disassembly tasks should ideally involve human–robot collaboration to leverage the strengths of both. The combination

Table 4
Rules for determining disassembly method.

No.	SWRL/SQWRL	Description
49	*fp* (?fp) ^*hDA* (?fp, ?a) ^*hDTl* (?fp, ?t) ^sqwrl:count (?a) = 1 ^sqwrl:count (?t) <= 1 → hDM (?fp, ?r) ^Robot (?r) ^hasProcessTime (?fp, ?pt)	For any *fp*, *ap* or *f* that has a single *DA* that uses less than or equal to one *DTl*, then the *fp*, *ap* or *f* can be disassembled by robot or human through certain process time
50	*fp* (?fp) ^*hDA* (?fp, ?a) ^*hDTl* (?fp, ?t) ^sqwrl:count (?a) = 1 ^sqwrl:count (?t) <= 1 → hDM (?fp, ?h) ^Human (?h) ^hasProcessTime (?fp, ?pt)	
51	*ap* (?ap) ^*hDA* (?ap, ?a) ^*hDTl* (?ap, ?t) ^sqwrl:count (?a) = 1 ^sqwrl:count (?t) <= 1 → hDM (?ap, ?r) ^Robot (?r) ^hasProcessTime (?ap, ?pt)	
52	*ap* (?ap) ^*hDA* (?ap, ?a) ^*hDTl* (?ap, ?t) ^sqwrl:count (?a) = 1 ^sqwrl:count(?t) <= 1 → hDM (?ap, ?h) ^Human (?h) ^hasProcessTime (?ap, ?pt)	
53	*f* (?f) ^*hDA* (?f, ?a) ^*hDTl* (?f, ?t) ^sqwrl:count (?a) = 1 ^sqwrl:count(?t) <= 1 → hDM (?f, ?r) ^Robot(?r) ^hasProcessTime (?f, ?pt)	
54	*f* (?f) ^*hDA* (?f, ?a) ^*hDTl* (?f, ?t) ^sqwrl:count (?a) = 1 ^sqwrl:count(?t) <= 1 → hDM (?f, ?h) ^Human(?h) ^hasProcessTime (?f, ?pt)	
55	*fp* (?fp) ^*hDA* (?fp, ?a) ^*hDTl* (?fp, ?t) ^sqwrl:count (?a) = 2 ^sqwrl:count(?t) <= 1 → hDM (?fp, ?r) ^Robot(?r) ^hasProcessTime (?fp, ?pt)	For any *fp*, *ap* or *f* that has two *DA*s that uses less than or equal to one *DTl*, then the *fp*, *ap* or *f* can be disassembled by robot through certain process time
56	*ap* (?ap) ^*hDA* (?ap, ?a) ^*hDTl* (?fp, ?t) ^sqwrl:count (?a) = 2 ^sqwrl:count (?t) <= 1 → hDM (?ap, ?r) ^Robot (?r) ^hasProcessTime (?ap, ?pt)	
57	*f* (?f) ^*hDA* (?f, ?a) ^*hDTl* (?fp, ?t) ^sqwrl:count (?a) = 2 ^sqwrl:count (?t) <= 1 → hDM (?f, ?r) ^Robot (?r) ^hasProcessTime (?f, ?pt)	
58	*fp* (?fp) ^*hDA* (?fp, ?a) ^*hDTl* (?fp, ?t) ^sqwrl:count (?a) = 2 ^sqwrl:count (?t) = 2 → hDM (?fp, ?h) ^Human (?h) ^hasProcessTime (?fp, ?pt)	For any *fp*, *ap* or *f* that has two *DA*s that uses two *DTl*s, then the *fp*, *ap* or *f* can only be disassembled by human through certain process time
59	*ap* (?ap) ^*hDA* (?ap, ?a) ^*hDTl* (?ap, ?t) ^sqwrl:count (?a) = 2 ^sqwrl:count (?t) = 2 → hDM (?ap, ?h) ^Human (?h) ^hasProcessTime (?ap, ?pt)	
60	*f* (?f) ^*hDA* (?f, ?a) ^*hDTl* (?f, ?t) ^sqwrl:count (?a) = 2 ^sqwrl:count(?t) = 2 → hDM (?f, ?h) ^Human (?h) ^hasProcessTime (?f, ?pt)	
61	*fp* (?fp) ^*hDA* (?fp, ?a) ^*hDTl* (?fp, ?t) ^sqwrl:count (?a) >2 ^sqwrl:count (?t) >= 2 → hDM (?fp, ?hr) ^HumanRobot (?hr) ^hasProcessTime (?fp, ?pt)	For any *fp*, *ap* or *f* that has more than two *DA*s that uses two and more than two *DTl*s, then the *fp*, *ap* or *f* can be disassembled by human–robot collaboration through certain process time
62	*ap* (?ap) ^*hDA* (?ap, ?a) ^*hDTl* (?ap, ?t) ^sqwrl:count (?a) >2 ^sqwrl:count (?t) >= 2 → hDM (?ap, ?hr) ^HumanRobot (?hr) ^hasProcessTime (?ap, ?pt)	
63	*f* (?f) ^*hDA* (?f, ?a) ^*hDTl* (?f, ?t) ^sqwrl:count (?a) >2 ^sqwrl:count (?t) >= 2 → hDM (?f, ?hr) ^HumanRobot (?hr) ^hasProcessTime (?f, ?pt)	

of human flexibility and robot precision appears to create a synergy that can handle complex disassembly tasks more effectively than either humans or robots working alone.

In the context of human–robot collaborative disassembly, existing knowledge does not allow for the determination of disassembly methods for various components due to the lack of a standard method for assessing the complexity of each task. Therefore, drawing on assessing criteria from the literature [60], we consider the number and type of disassembly actions and disassembly tools as criteria for determining the optimal disassembly methods for each task. The disassembly actions consist of nine basic actions adopted from literature [61], and the disassembly tools include common ones such as wrenches and screwdrivers, as well as specialised tools like bearing pullers. Building on this, this section introduces rules that use disassembly-related knowledge (required disassembly action and tool) to ascertain the optimal disassembly method for each component of an EoL product. The proposed rules and their descriptions are presented in Table 4. The optional disassembly methods for a component mainly include four main categories: executable by either humans or robots, solely by humans, exclusively by robots, or through human–robot collaboration.

- Rules 49–54 indicate that if a *c* can be disassembled through one *DA*, with less than or equal to one *DTl*, then the *c* can be disassembled either by a human or a robot.
- Rules 55–57, indicate that if a *c* can be disassembled through two *DA*s, with less than or equal to one *DTl*, then the *c* should be disassembled by a robot.
- Rules 58–60, indicate that if a *c* can be disassembled through two *DA*s, with two *DTl*s, then the *c* should be disassembled by a human.
- Rules 61–63, indicate that if a *c* can be disassembled through more than two *DA*s, with greater than or equal to two *DTl*s, then the *c* should be disassembled by human–robot collaborative.

Once the disassembly methods for each component have been determined through reasoning, the required disassembly time costs are integrated as additional knowledge.

4.3. Supportive rules

In addition to the rules previously established, there are several supportive rules proposed to specify and infer the optimal disassembly sequence as shown in Table 5.

- Rules 64–66 are utilised to update the available disassembly direction for each component.
- Rules 67–69, based on the direction and method of disassembly of each component and ensuring no precedence constraints, ascertain whether two components can be disassembled concurrently. Two components can only be simultaneously disassembled if they are oriented in different disassembly directions and use distinct disassembly methods, excluding the human–robot collaborative mode. Components of the same type typically employ identical disassembly methods; hence, simultaneous disassembly is not considered. Furthermore, the rules aim to minimise alterations in the disassembly direction and to ensure continuity in the disassembly actions.
- Rules 70–74 indicate that if a *f* has a hTBDDA relationship with another *fp* or *ap*, the *f* and *fp*/*ap* can be regarded as a *sa*. Similarly, if an *ap* has the hTBDDA relationship with another *ap*, these two *ap*s can be regarded as a *sa*. These rules can group components and simplify the disassembly sequence planning from reducing disassembly direction change, making the rule-based reasoning method more capable and efficient for larger and more complex EoL products.

In this study, the categorisation of subassemblies is contingent upon the types of components they contain. If a *sa* solely consists of a single type of *fp*, *ap*, or *f*, it can be regarded as that specific component type. Conversely, when a *sa* includes two or more types, it is defined as a *fp*.

Table 5
Auxiliary rules.

No.	SWRL/SQWRL	Description
64	*fp* (?fp) → dD_plusX (?fp, 1) ˆdD_minusX (?fp, 1) ˆ dD_plusY (?fp, 1) ˆdD_minusY (?fp, 1) ˆ dD_plusZ (?fp, 1) ˆdD_minusZ (?fp, 1)	Update and get the disassemble direction of a *fp*, *ap* or *f*.
65	*ap* (?ap) → dD_plusX (?ap, 1) ˆdD_minusX (?ap, 1) ˆ dD_plusY (?ap, 1) ˆdD_minusY (?ap, 1) ˆ dD_plusZ (?ap, 1) ˆdD_minusZ (?ap, 1)	
66	*f* (?f) → dD_plusX (?f, 1) ˆdD_minusX (?f, 1) ˆ dD_plusY (?f, 1) ˆdD_minusY (?f, 1) ˆ dD_plusZ (?f, 1) ˆdD_minusZ (?f, 1)	
67	*fp* (?fp) ˆ*ap* (?ap) ˆhDD (?fp, ?d1) ˆhDD (?ap, ?d2) ˆ swrlb:dF (?d1, ?d2) ˆhDM (?fp, ?m1) ˆhDM (?ap, ?m2) ˆ swrlb:dF (?m1, ?m2) ˆsqwrl:not (hTBDA (?fp, ?ap)) ˆ sqwrl:not (hTBDA (?ap, ?fp)) → cBDS (?fp, ?ap)	Determine if the *fp*/*ap*, *fp*/*f* or *ap*/*f* have different disassembly directions, can be disassembled by different method, and have no specified requirement for one part to be disassembled after the other. If these conditions are met, the rule infers that the referred parts can be disassembled simultaneously
68	*fp* (?fp) ˆ*f* (?f) ˆhDD (?fp, ?d1) ˆhDD (?f, ?d2) ˆ swrlb:dF (?d1, ?d2) ˆhDM (?fp, ?m1) ˆhDM (?f, ?m2) ˆ swrlb:dF (?m1, ?m2) ˆsqwrl:not (hTBDA (?fp, ?f)) ˆ sqwrl:not (hTBDA (?f, ?fp)) → cBDS (?fp, ?f)	
69	*ap* (?ap) ˆ*f* (?f) ˆhDD (?ap, ?d1) ˆhDD (?f, ?d2) ˆ swrlb:dF (?d1, ?d2) ˆhDM (?ap, ?m1) ˆhDM (?f, ?m2) ˆ swrlb:dF (?m1, ?m2) ˆsqwrl:not (hTBDA (?ap, ?f)) ˆ sqwrl:not (hTBDA (?f, ?ap)) → cBDS (?ap, ?f)	
70	hTBDDA (?f, ?fp) → *sa* (?f) ˆ*sa* (?fp)	If a *f* has to be direct disassembled after a *fp* or *ap*, the *fp* or *ap* has to be direct disassembled after the *f*. *Ap* A has to be direct disassembled after *ap* B. Then the *f* and *f*/*ap* can be regarded as a *sa*
71	hTBDDA (?fp, ?f) → *sa* (?fp) ˆ*sa* (?f)	
72	hTBDDA (?f, ?ap) → *sa* (?f) ˆ*sa* (?ap)	
73	hTBDDA (?ap, ?f) → *sa* (?ap) ˆ*sa* (?f)	
74	hTBDDA (?ap1, ?ap2) → *sa* (?ap1) ˆ*sa* (?ap2)	

4.4. Process for generating the optional disassembly scheme

Based on the constructed product semantic model, the research workflow of this study primarily encompasses two segments: the establishment of disassembly precedence constraints among product components and the determination of their disassembly methods, followed by the generation of the optimal disassembly scheme.

4.4.1. Workflow for generating disassembly precedence constraints

The workflow for generating disassembly precedence constraints among components is illustrated in Fig. 6. Prior to executing the relevant reasoning rules, three empty sets are initialised, namely Set *f*, Set *fp*, and Set *ap*. In this study, the primary input comprises the lowest-level components of the EoL product. Different components, according to the categories, are assigned to different sets. Disassembly-related information of components can be stored and transmitted in different sets. After reading and inputting the complete product components, rules 1–48 are utilised to establish the disassembly precedence constraints among the various components. The constructed disassembly precedence constraints, built upon the foundation of traditional And/Or graphs, retain additional information about the components, thereby facilitating subsequent decision-making processes. Subsequently, based on rules 70–74, related components are grouped together and treated as subassemblies, reducing the overall number of product components and thereby diminishing the solution space of feasible disassembly sequences. The types of sub-components are determined based on the categories of components they contain, while concurrently updating the components in each set. Finally, using rules 49–63, the potential disassembly methods for each component are inferred. The time cost associated with each disassembly method is stored as known knowledge.

4.4.2. Workflow for generating optimal disassembly scheme

Based on the constructed precedence constraints through Section 4.4.1, all optional disassembly sequence and optimal disassembly scheme can be generated and determined through using subsequent rules. The overall workflow to determine the optimal disassembly scheme is illustrated in Fig. 7. According to the grouping and determining process in Section 4.4.1, the *sas* are already sequentially grouped which can be categorised and regarded as *fps*, *aps* or *f*s. Then, dividing all components into three different sets *fp*, *ap* and *f* according to the attribute of each component of a EoL product. It is worth noting that the relationship of a *f* is direct covered by another *fp* or *ap* will be eliminated through the grouping and generating of *sas*. This procedure not only reduced the number of disassembly components in sequence planning, but also complied with the normal consistency of disassembly practice.

After applying rules 64–66, the available disassembly direction and all executable components are identified and initialised. Following conventional disassembly practices and loosening the connection of components, the *f*s are determined to be disassembled at first. Then, the set *f* is evaluated, if it is non-empty, one executable *f* is chosen as the current disassembly task. Simultaneously, the feasibility of simultaneous disassembly between this *f* and other executable *fp* or *ap* is determined through rules 68–69. If a *fp*/*ap* is identified, the *f* and the *fp*/*ap* are to be disassembled together. Otherwise, the *f* is to be disassembled individually. Upon the completion of this disassembly task, the disassembly directions for the remaining components are reinitialised using rules 64–66, and relevant sets are updated. The process iteratively continues.

When set *f* is empty, set *ap* is evaluated. Typically, *aps* are added to enhance the overall performance of the product that bridge the connection between *f*s and *fps*. If set *ap* is non-empty, an executable *ap* is selected as the current disassemble task. Rule 67 aids in assessing whether the *ap* can be disassembled concurrently with any remaining *fp*. If a *fp* is identified, they are disassembled together. Otherwise, the *ap* is disassembled individually. Once the *ap* has been disassembled, the available disassembly directions for the remaining *fps* or *aps* are updated according to rules 64–65, subsequently updating sets *fp* and *ap*.

When set *ap* is empty, set *fp* is finally examined. If set *fp* non-empty, an executable *fp* becomes the current task. Normally, these remaining *fps* possess a higher residual value and are disassembled using similar methods. Therefore, no further evaluation is conducted to identify the remaining *fps* that can be disassembled simultaneously. Iteratively, once the set *fp* is empty, it can be concluded that all components of

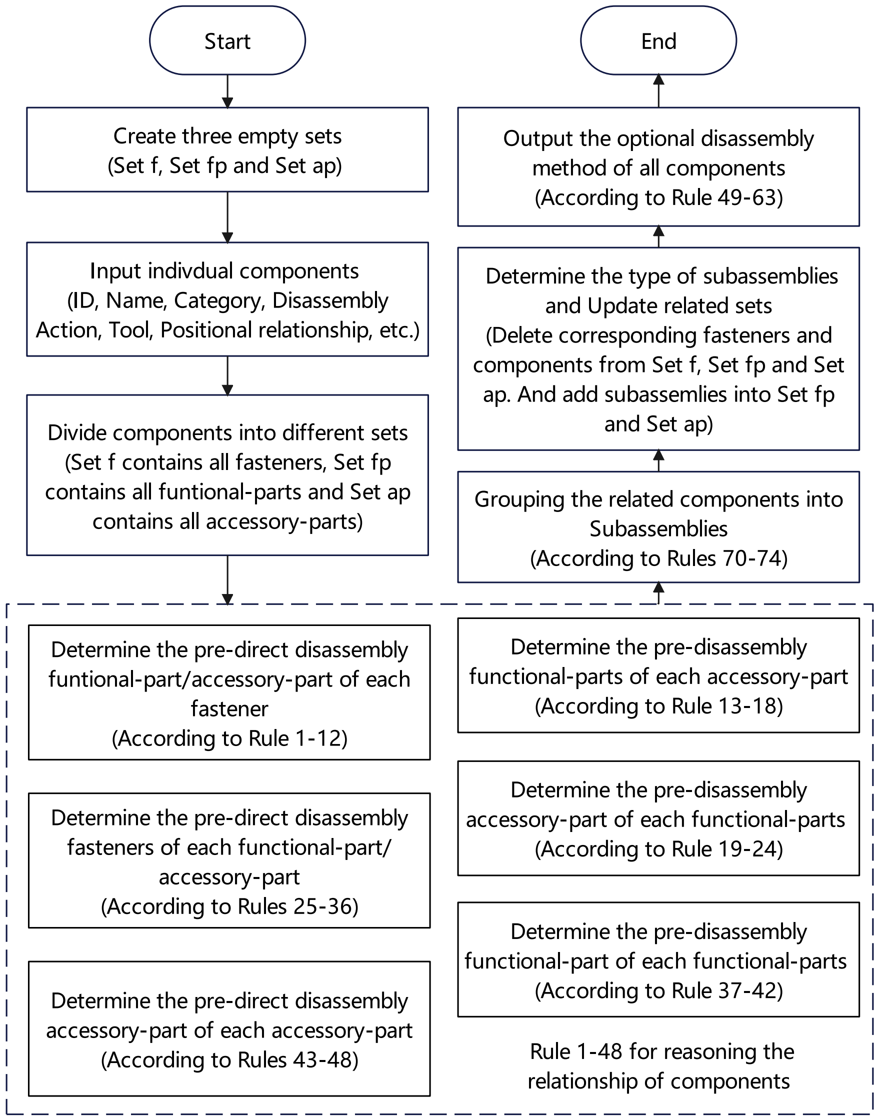


Fig. 6. Workflow for generating precedence constraints of EoL product.

the EoL product have been completely disassembled. Subsequently, all optional human–robot collaborative disassembly sequences are generated and outputted by sequentially following the disassembly tasks. The optimal human–robot collaborative disassembly scheme then can also be determined through the certain criteria.

In addition, in order to resolve the conflicts and uncertain of multi-optional simultaneous disassembly tasks, the selection of executable components is conducted in the order of +Y/+X/+Z/−X/−Z, with −Y typically representing the gravitational direction and not considered for executing disassembly tasks.

5. Case study

In this section, we use a worm gear reducer gearbox as a case study for human–robot collaborative disassembly sequence planning to validate the feasibility and efficiency of the proposed methods. The exploded view and bill of materials for the gearbox are illustrated in the respective Fig. 8 and Table 14. The CAD file and bill of material (BOM) can be downloaded and reached through the link in Appendix B. This case study serves to validate the proposed human–robot collaborative disassembly ontology and inference rules. In this case study, the gearbox is considered to be completely disassembled using a non-destructive disassembly type. The *ProcessingTime* of this case study is in dimensionless units.

5.1. Disassembly ontology model of gearbox

In this study, the disassembly ontology of the gearbox is constructed using the modelling tool Protégé 5.5.0. Based on the general disassembly ontology model developed, the components of the gearbox are input as instances. The established classes, objective properties, data properties and individuals of the gearbox are shown in Fig. 9. The created gearbox ontology is saved in the OWL Web Ontology Language format (.owl file). The rules pertinent to human–robot collaborative disassembly are developed using Microsoft Visual Studio Community 2019, utilising the C# programming language. This integration leverages the open-source .Net library (dotNetRDF), which offers a robust API for using SPARQL, along with Jena.Net, a flexible .NET port of the Jena semantic web toolkit. In this research, the ontology file generated and saved in Protégé (.owl file) is read and imported into Visual Studio. Hence, using SPARQL and the Jena Ontology API, the corresponding rule inferences are constructed and executed, facilitating the generation of the optimal human–robot collaborative disassembly scheme.

5.2. Precedence constraints of the gearbox

The semantic disassembly relationships of fasteners, accessory parts and functional parts in the gearbox are presented in Tables 6, 7 and

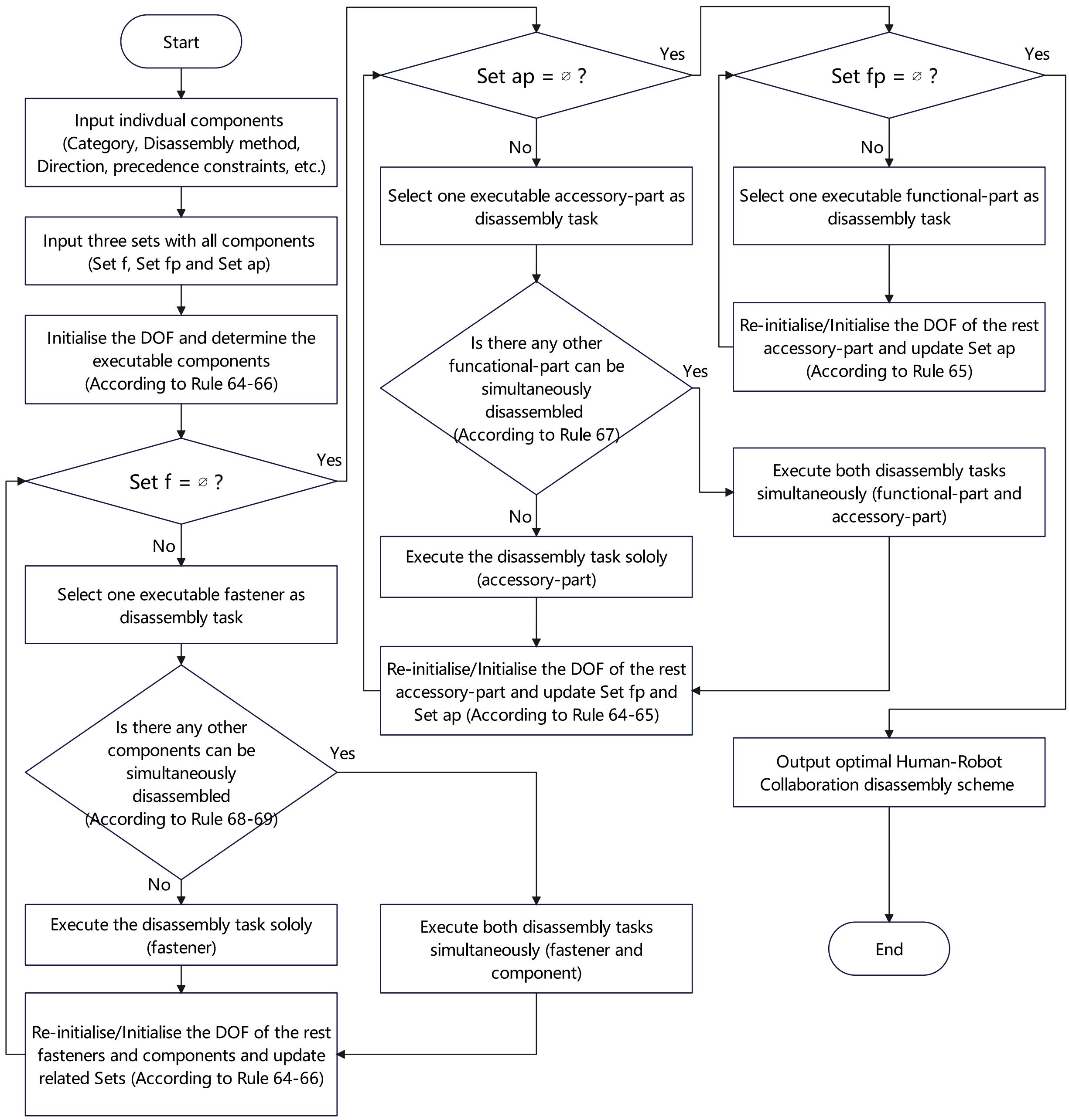


Fig. 7. Workflow for generating optimal disassembly schemes.

Table 6
Semantic assembly-relations of fasteners in the gearbox.

f No.	iDCB_plusX	iDCB_minusX	iDCB_plusY	iDCB_minusY	iDCB_plusZ	iDCB_minusZ
4	–	–	–	–	–	–
5	–	–	–	–	–	–
f No.	cBDI_plusX	cBDI_minusX	cBDI_plusY	cBDI_minusY	cBDI_plusZ	cBDI_minusZ
4	1	0	0	0	0	0
5	0	0	0	0	1	0

8, respectively. The precedence constraints of the gearbox, obtained by executing the workflow in Section 4.4.1, are illustrated in Fig. 10. The component disassembly precedence relationships, inferred through Rules 1–48, are showcased in Fig. 10(a). By implementing subsequent Rules 49–63 and 70–74, the simplified disassembly precedence relationships are presented in Fig. 10(b). According to the precedence constraint graphs, the precedence constraints are reduced from 22 to 7. The optional disassembly sequences are significantly reduced while

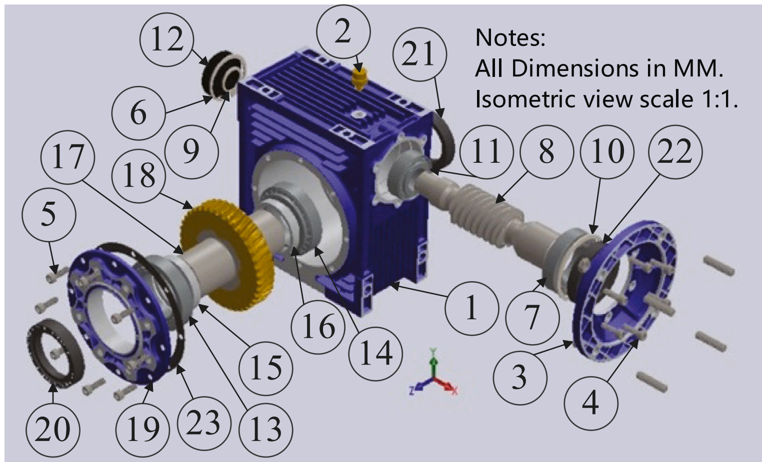


Fig. 8. Exploded view of gearbox.

Table 7
Semantic assembly-relations of accessory parts in the gearbox.

ap No.	iDCB_plusX	iDCB_minusX	iDCB_plusY	iDCB_minusY	iDCB_plusZ	iDCB_minusZ
6	–	12	–	–	–	–
9	–	6	–	–	–	–
10	22	–	–	–	–	–
15	–	–	–	–	13	–
16	–	–	–	–	–	14
19	–	–	–	–	20	–
20	–	–	–	–	–	–
21	–	–	–	–	–	–
22	3	–	–	–	–	–
23	–	–	–	–	19	–

ap No.	iFB_plusX	iFB_minusX	iFB_plusY	iFB_minusY	iFB_plusZ	iFB_minusZ
6	–	–	–	–	–	–
9	–	–	–	–	–	–
10	–	–	–	–	–	–
15	–	–	–	–	–	–
16	–	–	–	–	–	–
19	–	–	–	–	5	–
20	–	–	–	–	–	–
21	–	–	–	–	–	–
22	–	–	–	–	–	–
23	–	–	–	–	–	–

the effectiveness of planning procedure is improved through rules 70–74. The information results pertaining to the generated subassemblies and the remaining components are detailed in Table 9. Notably, five subassemblies were generated, with the S2 subassembly classified as an accessory part, while the rest are identified as functional parts.

5.3. Optimal human–robot collaborative disassembly scheme of the gearbox

In this section, we generate all feasible disassembly sequences and determine the optimal human–robot collaborative disassembly scheme for the gearbox using the methods proposed in this study. Additionally, we employ several fundamental heuristic optimisation algorithms as benchmarks to demonstrate the superiority of our approach.

5.3.1. Our approach

Based on the established simplified component precedence constraints, accessory part S2 was initially identified as the primary task for disassembly. By employing Rule 67, it was deduced that S2 could be disassembled concurrently with functional part 2. By executing the workflow in Section 4.4.2, two feasible disassembly solutions are generated: Solution 1 and 2, as shown in Table 10.

In Solution 1, the disassembly of accessory part S2 is performed by a human, while functional component 2 is disassembled by a robot. Under this schema, the resultant unit disassembly time amounts to 336 units. In contrast, Solution 2 envisions the robotic disassembly of accessory part S2, concurrent with the disassembly of functional component 2 by a human. Due to the configuration in Solution 2, where accessory part S2 is still being processed by the robot as the human completes disassembly task 2, the subsequent accessory part 21 can be disassembled by the human immediately after the completion of the task (functional component 2). This synchronisation further trims the overall disassembly time, resulting in a unit disassembly time of 308, marking this solution as optimal.

5.3.2. Comparison experiments

This research is the first to solve the human–robot collaborative disassembly sequence planning problem using an ontology model and a pure rule-based reasoning method. Additionally, by taking a gearbox as a case study, we generate the optimal human–robot collaborative disassembly scheme through the proposed method. Since there is no identical method to prove the superiority of our proposed method, we attempt to evaluate its feasibility and effectiveness by comparing it with representative optimisation algorithms.

According to the review paper [62], the state-of-the-art and the most common used methods for human–robot collaborative disassembly sequence planning and task allocation are genetic algorithm (GA), artificial bee colony (ABC), ant colony (AC), particle swarm optimisation (PSO), linear programming (LP) and Tabu search (TS). Due to LP

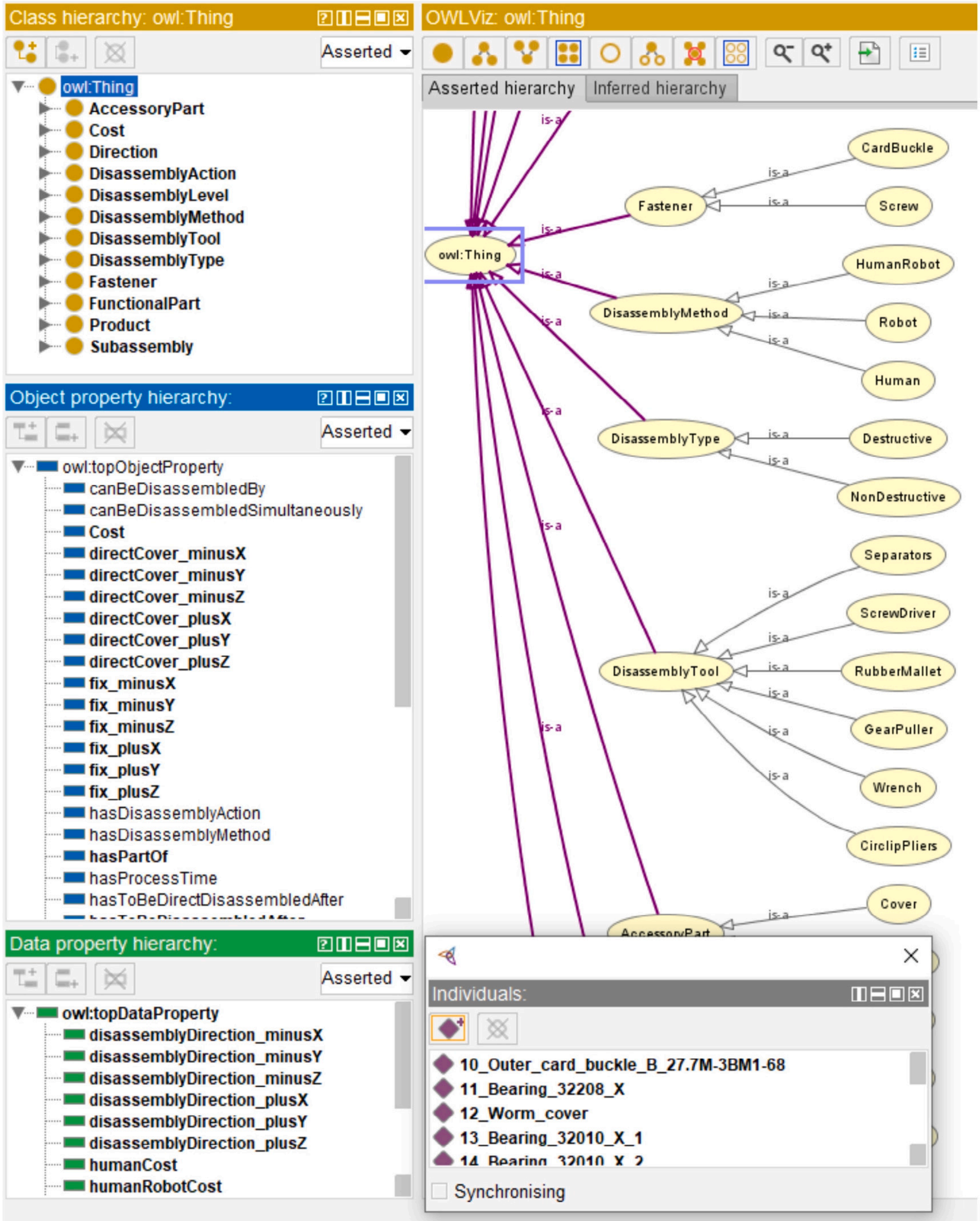


Fig. 9. Screenshot of the human–robot disassembly ontology of gearbox in Protégé.

requires formal mathematical model which is out of the research scope in this research, the other five representative algorithms are taken in this comparison experiment.

The disassembly-related information and optional disassembly methods in Table 14 are considered as inputs. The precedence constraints in Fig. 10 are input as restrictions. One disassembly task without any precedence constraints can be concurrently disassembled with another task through different directions and methods. Moreover, the subassembly has not been considered in these optimisation algorithms.

As mentioned above, the main purpose of this comparison experiment is to validate the feasibility and effectiveness of the proposed method. The design, parameter setting, and performance optimisation of these optimisation algorithms are beyond the scope of this study. Therefore, we employed five representative optimisation algorithms from an open-source library (the link is available in Appendix B), with relevant parameters set to their default values as provided by the library. These parameters are presented in Table 11.

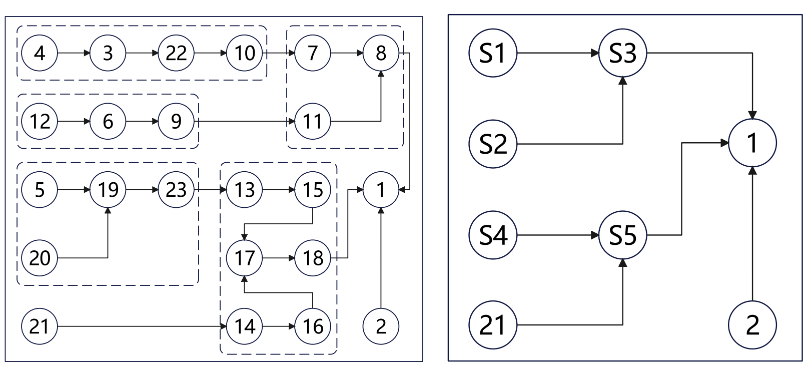
The number of iterations for this comparison experiment is set at 1000, which serves as one termination condition. Another termination condition occurs when a solution is found during these iterations and is not updated in the subsequent four iterations. At that point, the iterations will stop, and the current solution will be considered the optimal solution. To ensure the fairness and effectiveness of these optimisation algorithms, we have conducted 20 experiments under

Table 8
Semantic assembly-relations of functional parts in the gearbox.

fp No.	iDCB_plusX	iDCB_minusX	iDCB_plusY	iDCB_minusY	iDCB_plusZ	iDCB_minusZ
1	8	–	2	–	18	21
2	–	–	–	–	–	–
3	–	–	–	–	–	–
7	10	–	–	–	–	–
8	7	11	–	–	–	–
11	–	9	–	–	–	–
12	–	–	–	–	–	–
13	–	–	–	–	23	–
14	–	–	–	–	–	21
17	–	–	–	–	15	16
18	–	–	–	–	17	–
fp No.	iFB_plusX	iFB_minusX	iFB_plusY	iFB_minusY	iFB_plusZ	iFB_minusZ
1	–	–	–	–	–	–
2	–	–	–	–	–	–
3	4	–	–	–	–	–
7	–	–	–	–	–	–
8	–	–	–	–	–	–
11	–	–	–	–	–	–
12	–	–	–	–	–	–
13	–	–	–	–	–	–
14	–	–	–	–	–	–
17	–	–	–	–	–	–
18	–	–	–	–	–	–

Table 9
Output of optional disassembly methods of components in the gearbox.

No.	Components	Quantity	Category	*DA*	*DTl*	*DM*	ProcessTime
S1	3,4,10,22	9	*fP*	Grasp, unscrew, unplug	Screwdriver, puller, circlip pliers	H	34
S2	6,9,12	3	*ap*	Unplug	Circlip pliers	H/R	39/26
S3	7,8,11	3	*fp*	Place, grasp, move, unplug, slide, rotate	Puller, separators, circlip pliers	HR	86
S4	5,19,20,23	9	*fp*	Unscrew, move, grasp, unplug	Screwdriver, puller, rubber mallet, circlip pliers	HR	42
S5	13–18	6	*fp*	Place, grasp, move, slide, unplug, rotate	Puller, separators, circlip pliers	HR	120
2	–	1	*fp*	Rotate	Wrench	H/R	10/8
21	–	1	*ap*	Grasp, unplug	puller, circlip plier	H	15



(a) Initial reasoning result. (b) Simplified reasoning result.

Fig. 10. Precedence constraints of the gearbox.

Table 10
Optional human–robot collaborative disassembly sequence schemes.

No.	Disassembly sequence	*DM*	ProcessTime
1	⟨S2(−X), 2(−Y)⟩ →21(−Z)→S1(+X)→S3(+X)→S4(+Y)→S5(+Y)	⟨H, R⟩→H→HR→HR→HR→HR	336
2	⟨S2(−X), [2(+Y), 21(−Z)]⟩→S1(+X)→S3(+X)→S4(+Y)→S5(+Y)	⟨R, [H]⟩→H→HR→HR→HR→HR	**308**

the same environment. The optimal experiment results are shown in Table 12.

In the table provided, we present the results of disassembly sequences derived from the chosen five representative optimisation algorithms. The optimisation outcomes for *ProcessTime* are similar, ranging from 310 to 320 units. The TS emerges as the most efficient, with a process time of 311 units, which is the closest to our optimal result. The GA, ABC, and AC share similar process times of 313 units, 314 units, and 315 units, respectively. In contrast, the PSO requires the most process time, consuming 319 units and marking a slight delay compared to the others. Notably, none of the representative optimisation algorithms can achieve the optimal process time of 308 units from our method.

In this comparison experiment, we do not set any preference and restriction on the disassembly direction. However, different algorithms suggest various initial disassembly directions. For instance, TS, ABC, and GA start with the −X direction for the same task in disassembly,

Table 11
Parameter settings for comparison experiment.

Method	Parameters	Values	Method	Parameters	Values
TS	Number of iterations	1000	ABC	Population size	50
	Number of neighbours	10		Number of iterations	1000
	Tabu size	10		N_limits	25
GA	Population size	50	AC	Population size	50
	Number of iterations	1000		Number of iterations	1000
	Crossover Probability	0.9		Sample count	25
	Mutation Probability	0.1		Intent factor	0.5
PSO	Number of iterations	1000		Deviation distance ratio	1.0
	Local coefficient	2.05			
	Global coefficient	2.05			
	Inertia factor	0.4			

while PSO and AC begin with the +Y direction and +Z direction, respectively. All algorithms result in more than 10 direction changes through the disassembly process. Frequent direction changes, which can increase time and labour costs and raise safety issues, are not ideal in practical disassembly processes. In contrast, guided by expert knowledge and disassembly preferences, our method tends to perform successive operations in consistent directions, thereby minimising directional changes in the disassembly process, which requires only 3 direction changes.

Indeed, the representative optimisation algorithms have certain limitations due to their default parameter settings. They require further fine-tuning of the involved parameters to achieve better solutions towards the global optimum. Without support from expert knowledge and real-world scenarios, the fine-tuning process can be complex and unpredictable. According to the comparison experiment, the significant advantages of our method are summarised in Table 13.

6. Discussion

This paper proposes a human–robot disassembly ontology model and a rule-based reasoning method to determine the optimal disassembly scheme of EoL products. Compared with the proposed methods constructed in other studies [48,49], the approach in this research has the following innovative points:

1. For the first time, a general disassembly ontological semantic model for a human–robot collaborative disassembly environment has been developed. The method proposed in this study has general applicability. Based on the human–robot collaborative disassembly ontology, any mechanical product or disassembly case can be semantically represented using structured language. Given the inclusion of product-specific information, it represents high flexibility and efficiency.
2. The categories of product components have been further developed. In existing methods, product components are typically divided into functional parts and fasteners. This classification facilitates the establishment of product disassembly priority constraints based on product topology. However, this classification is too general, making it challenging to retrieve relevant information about the components themselves. Thus, in this research, considering the scenario of human–robot collaborative disassembly in remanufacturing, a three-category division was introduced: functional parts, accessory parts, and fasteners. Although this elevates the complexity of constructing product topology and disassembly priority constraints, it facilitates information retrieval of relevant components and provides decision-making information for subsequent remanufacturing steps.
3. Different from typical heuristic optimisation algorithms, this study uses purely semantically constructed disassembly rules to infer and determine the optimal human–robot collaborative disassembly sequence and scheme. These disassembly rules can be represented using SWRL/SQWRL, enabling computers to interact with robots. Moreover, feasible disassembly sequences are generated by effectively reasoning through these rules, thereby retaining high-quality executable disassembly plans that meet the demands of complex mechanical devices. Additionally, the rules constructed based on semantic principles demonstrate excellent flexibility and efficiency.

However, the method proposed in this study have some limitations:

1. In this study, the overall process time and number of disassembly direction change are considered as two objectives for optimisation in human–robot collaborative disassembly sequence planning. The process time is considered a representation of the cost. However, in real-world and practical scenarios, this research does not consider some other factors, such as human labour costs, robot operating costs, and the safety of human–robot collaboration, which significantly influence the allocation of disassembly tasks.
2. The system-level ontology model and rule-based reasoning method proposed in this study are at an initial stage. They focus solely on generating the precedence constraints and the optimal disassembly scheme for EoL products under specified conditions. Operational level factors, such as unpredictability of humans and robots, uncertainties and faults in EoL products, and safety concerns, have not been considered in this research.
3. In this study, only one type of gearbox is used as a case study to validate the proposed method. It is possible to integrate our proposed method into the human–robot disassembly sequence planning for all EoL mechanical product. However, due to the limited number of study cases, an effective product case library has yet to be established.

7. Conclusions and future work

Human–robot collaborative disassembly, as an emerging semi-automatic remanufacturing paradigm, can effectively promote the automation of the disassembly process. However, due to the availability of multiple disassembly methods, planning and determining the optimal scheme for human–robot collaborative disassembly becomes more difficult and complex. Currently, most planning and optimisation methods for disassembly rely on heuristic algorithms. The operating mechanism of a heuristic algorithm makes it difficult to monitor each disassembly step, and its solutions cannot be definitively determined as globally optimal.

In this context, we proposed a human–robot collaborative disassembly ontology model and rule-based reasoning method to plan and determine the optimal disassembly scheme in remanufacturing. A general human–robot collaborative disassembly ontology model is proposed to organise and formalise the disassembly-related information of EoL products. This model constructs a standardised and semantically structured representation of disassembly precedence constraints, along with optional disassembly methods, for each disassembly component of EoL products. Additionally, by establishing related semantic rules, the optimal disassembly scheme for EoL products is inferred, generated, and determined step by step in a more clear and transparent manner. The feasibility of the proposed method is validated through a case study and comparison experiments. When compared to six other basic optimisation algorithms, our method achieves the shortest process time of 308 units and the fewest number of direction changes of 3 times. Our method facilitates the integration, sharing, and expansion of disassembly knowledge, ultimately offering a flexible method for determining disassembly solutions for various EoL products. For future research, the following three primary aspects should be considered:

Table 12
Comparison experiment results.

No.	Disassembly sequence	Direction change times	ProcessTime
TS	12(−X)→21(−Z)→5(+Z)→14(−Z)→20(+Z)→16(−Z)→ 4(+X)→6(−X)→19(+Z)→3(+X)→22(+X)→9(−X)→ 23(+Z)→10(+X)→11(−X)→13(+Z)→15(+Z)→2(+Y)→ 17(+Z)→7(+X)→18(+Z) →8(+X)	19	311
ABC	12(−X)→20(+Z)→6(−X)→4(+X)→5(+Z)→3(+X)→ 22(+X)→19(+Z)→23(+Z)→9(−X)→11(−X)→13(+Z)→ 15(+Z)→21(−Z)→14(−Z)→16(−Z)→17(+Z)→18(+Z)→ 2(+Y)→10(+X)→7(+X)→8(+X)	12	314
AC	20(+Z)→12(−X)→6(−X)→4(+X)→21(−Z)→3(+X)→ 9(−X)→2(+Y)→14(−Z)→5(+Z)→19(+Z)→22(+X)→ 10(+X)→7(+X)→11(−X)→8(+X)→16(−Z)→23(+Z)→ 13(+Z)→15(+Z)→17(+Z)→18(+Z)	13	315
PSO	2(+Y)→4(+X)→3(+X)→21(−Z)→12(−X)→22(+X)→ 10(+X)→6(−X)→20(+Z)→9(−X)→11(−X)→7(+X)→ 8(+X)→14(−Z)→5(+Z)→19(+Z)→16(−Z)→23(+Z)→ 13(+Z)→15(+Z)→17(+Z)→18(+Z)	12	319
GA	12(−X)→4(+X)→2(+Y)→5(+Z)→3(+X)→20(+Z)→ 19(+Z)→6(−X)→9(−X)→21(−Z)→11(−X)→14(−Z)→ 8(+X)→13(+Z)→15(+Z)→22(+X)→10(+X)→7(+X)→ 23(+Z)→16(−Z)→17(+Z)→18(+Z)	15	313
Ours	{S2(−X), [2(+Y), 21(−Z)]}→S1(+X)→S3(+X)→S4(+Y)→S5(+Y)	**3**	**308**

Table 13
Comparison between our method and optimisation algorithms..

	Our method (OM)	Optimisation algorithms (OAs)
Principle	OM is proposed based on SWRL rules, which are a formal semantic expression derived from expert knowledge and real- world scenarios.	OAs are proposed based on a defined objective function and are subject to various constraints, but they lack expert knowledge and descriptions of real-world scenarios.
Procedure	Transparent and explicit.	Opaque
Predictability	OM reasoning outcomes are predictable. Outcomes are consistent as long as the input conditions are the same.	The results of OAs are unpredictable due to the randomness of OAs.
Traceability	OM is easy to trace back to specific rules.	OAs act as a black box, and the intermediate processes are hard to trace.
Complexity	The semantic expressions and SWRL rules in OM are easier to implement and understand.	OAs are more complex to implement, which require algorithm design and constant adjustment of parameters.
Replicability	The SWRL rules in OM can directly encode, store, and integrate the expert knowledge into different scenarios.	OAs are designed, and parameters are adjusted to a specific scenario.
Optimisation result	OM can generate an exact optimal solution.	OAs generate a near-optimal solution and are unable to confirm that the solution is optimal.

Table 14
Disassembly-related information of the gearbox.

ID	Name	Instance in Protégé	Category	Quantity	*DA*	*DTl*	*DM*	ProcessTime
1	House	house	*fp*	1	–	–	–	–
2	Ventilator	ventilator	*fp*	1	Rotate	Wrench	H/R	10/8
3	Flange	flange	*fp*	1	Unscrew	–	H/R	15/10
4	Screw-1	screw-1	*f*	6	Unscrew	Screwdriver	H/R	5/2
5	Screw-2	screw-2	*f*	6	Unscrew	Screwdriver	H/R	5/2
6	Outer Buckle-52	outer_buckle_52	*ap*	1	Unplug	Circlip pliers	H/R	8/6
7	Bearing-32205	bearing_32205	*fp*	1	Place, grasp, move, unplug	Puller, separators, circlip pliers	HR	30
8	Worm Shaft	worm_shaft	*fp*	1	Grasp, move, slide, rotate	Puller, separators, circlip pliers	HR	26
9	Inner Buckle	inner_buckle	*ap*	1	Unplug	Circlip pliers	H/R	17/10
10	Outer Buckle-68	outer_buckle_68	*ap*	1	Unplug	Circlip pliers	H/R	12/9
11	Bearing-32008	bearing_32008	*fp*	1	Place, grasp, move, unplug	Puller, separators, circlip pliers	HR	30
12	Worm Cover	worm_cover	*ap*	1	Unplug	Circlip pliers	H/R	14/10
13	Bearing-32010–1	bearing_32010_1	*fp*	1	Place, grasp, move, unplug	Puller, separators, circlip pliers	HR	25
14	Bearing-32010–2	bearing_32010_2	*fp*	1	Place, grasp, move, unplug	Puller, separators, circlip pliers	HR	25
15	Gear spacer-1	gear_spacer_1	*fp*	1	Grasp, unplug	Circlip pliers	R	14
16	Gear spacer-2	gear_spacer_2	*ap*	1	Grasp, unplug	Circlip pliers	R	14
17	Gear shaft	gear_shaft	*fp*	1	Grasp, move, slide, rotate	Puller, separators, circlip pliers	HR	23
18	Gear	gear	*fp*	1	Slide, move, grasp	Puller, separators, circlip pliers	HR	45
19	Gear Cover	gear_cover	*ap*	1	Move	Rubber mallet	H/R	20/12
20	Gear retentor-1	gear_retentor_1	*ap*	1	Grasp, Unplug	Puller, Circlip plier	H	15
21	Gear retentor-2	gear_retentor_2	*ap*	1	Grasp, Unplug	Puller, circlip plier	H	15
22	Worm retentor	worm_retentor	*ap*	1	Grasp, Unplug	Puller, circlip plier	H	22
23	Cover retentor	cover_retentor	*ap*	1	Grasp, Unplug	Puller, circlip plier	H	18

1. This study is the first to introduce an ontology model and a rule-based reasoning method for human–robot collaborative disassembly sequence planning. Therefore, this research focuses on proposing the upper-layer framework of the ontology model and the reasoning rules in human–robot collaborative disassembly. Practical factors such as uncertainty and failure are not considered in this research. The ontology model can be further expanded through integrating these factors to reflect a more practical industrial scenario. Moreover, the generative pre-trained transformer (GPT) models offer a potential solution to generate, learn and update the ontology model automatically.
2. In this research, the generated disassembly strategies consider only a complete and damage-free disassembly mode. Feasible and optimal disassembly schemes are determined based on the overall process time and the number of disassembly direction changes. However, due to the various constraints and uncertainties associated with product components in real-world disassembly, as well as potential failures during the disassembly process, the complexity evaluation of each disassembly task should be enhanced by considering economic and technical factors. Subsequently, the selection of disassembly methods and the optimisation of human–robot collaborative disassembly schemes will ensure greater practical significance and value.
3. The proposed ontology model and disassembly-related rules can be further adapted and expanded to encompass the entire remanufacturing process [63]. The full life-cycle of EoL products can be incorporated into the EoL product ontology knowledge base, which can also aid in establishing the digital twin model of EoL products. Furthermore, the SWRL rules have the potential to support and enhance planning and optimisation within the entire remanufacturing process. Consequently, a smart remanufacturing system embedded with the ontology model and rule-based reasoning mechanism can be established.

CRediT authorship contribution statement

Youxi Hu: Conceptualisation, Methodology, Software, Validation, Investigation, Data curation, Writing – original draft, Visualisation. **Chao Liu:** Methodology, Formal analysis, Supervision, Writing—review & editing. **Ming Zhang:** Conceptualisation, Formal analysis. **Yuqian Lu:** Methodology, Writing—review & editing. **Yu Jia:** Methodology, Formal analysis, Writing—review & editing. **Yuchun Xu:** Methodology, Resources, Funding acquisition, Supervision, Project administration, Writing—review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The example greabox and its data used in our case study is open source and we have provided a link to the website in our manuscript.

Acknowledgements

This research work was funded by RECLAIM project 'Remanufacturing and Refurbishment of Large Industrial Equipment' and received funding from the European Commission Horizon 2020 research and innovation programme under Grant Agreement No. 869884. This research work was partially supported by the National Natural Science Foundation of China (No. 52105534).

Appendix A. Disassembly-related information of the gearbox

The BOM and related information of the gearbox are presented in Table 14.

Appendix B. Link for the illustrative example and the case study

Open source:
https://www.youtube.com/watch?v=b6h_ZiGoLY0
https://grabcad.com/library/worm-gear-reducer-13
https://grabcad.com/library/belt-roller-support-assembly-in-solidworks-1
https://github.com/thieu1995/mealpy

References

[1] O. Okorie, F. Charnley, A. Ehiagwina, D. Tiwari, K. Salonitis, Towards a simulation-based understanding of smart remanufacturing operations: a comparative analysis, J. Remanuf. (2020) 1–24.
[2] R.F. Fofou, Z. Jiang, Y. Wang, A review on the lifecycle strategies enhancing remanufacturing, Appl. Sci. 11 (13) (2021) 5937.
[3] A. Zacharaki, T. Vafeiadis, N. Kolokas, A. Vaxevani, Y. Xu, M. Peschl, D. Ioannidis, D. Tzovaras, RECLAIM: Toward a new era of refurbishment and remanufacturing of industrial equipment, Front. Artif. Intell. 3 (2021) 570562.
[4] Y. Hu, C. Liu, M. Zhang, Y. Jia, Y. Xu, A novel simulated annealing-based hyper-heuristic algorithm for stochastic parallel disassembly line balancing in smart remanufacturing, Sensors 23 (3) (2023) 1652.
[5] Y. Hu, M. Zhang, C. Liu, Y. Xu, Z. Wang, Y. Jia, A simulated annealing hyper-heuristic algorithm for process planning and scheduling in remanufacturing, in: 2022 27th International Conference on Automation and Computing, ICAC, IEEE, 2022, pp. 1–6.
[6] M. Zhang, Y. Lu, Y. Hu, N. Amaitik, Y. Xu, Dynamic scheduling method for job-shop manufacturing systems by deep reinforcement learning with proximal policy optimization, Sustainability 14 (9) (2022) 5177.
[7] C.-M. Lee, W.-S. Woo, Y.-H. Roh, Remanufacturing: Trends and issues, Int. J. Precis. Eng. Manuf. Green Technol. 4 (1) (2017) 113–125.
[8] M. Kerin, D.T. Pham, A review of emerging industry 4.0 technologies in remanufacturing, J. Clean. Prod. 237 (2019) 117805.
[9] S. El Kadiri, D. Kiritsis, Ontologies in the context of product lifecycle management: state of the art literature review, Int. J. Prod. Res. 53 (18) (2015) 5657–5668.
[10] A.K. Inkulu, M.V.A.R. Bahubalendruni, D. Ashok, K. SankaranarayanaSamy, Challenges and opportunities in human robot collaboration context of industry 4.0 - a state of the art review, Ind. Robot. 49 (2) (2022) 226–239.
[11] K. Keung, C. Lee, X. Liqiao, L. Chao, L. Bufan, P. Ji, A cyber-physical robotic mobile fulfillment system in smart manufacturing: The simulation aspect, Robot. Comput.-Integr. Manuf. 83 (2023) 102578.
[12] J. Huang, D.T. Pham, Y.J. Wang, C.Q. Ji, W.J. Xu, Q. Liu, Z.D. Zhou, A strategy for human-robot collaboration in taking products apart for remanufacture, FME Trans. 47 (4) (2019) 731–738.
[13] H. Wang, J. Tao, T. Peng, A. Brintrup, E.E. Kosasih, Y. Lu, R. Tang, L. Hu, Dynamic inventory replenishment strategy for aerospace manufacturing supply chain: combining reinforcement learning and multi-agent simulation, Int. J. Prod. Res. 60 (13) (2022) 4117–4136.
[14] Z. Liu, Q. Liu, W. Xu, L. Wang, Z. Zhou, Robot learning towards smart robotic manufacturing: A review, Robot. Comput.-Integr. Manuf. 77 (2022) 102360.
[15] D. Mukherjee, K. Gupta, L.H. Chang, H. Najjaran, A survey of robot learning strategies for human-robot collaboration in industrial settings, Robot. Comput.-Integr. Manuf. 73 (2022) 102231.
[16] R. Maderna, M. Pozzi, A.M. Zanchettin, P. Rocco, D. Prattichizzo, Flexible scheduling and tactile communication for human–robot collaboration, Robot. Comput.-Integr. Manuf. 73 (2022) 102233.
[17] C. Chen, C. Liu, T. Wang, A. Zhang, W. Wu, L. Cheng, Compound fault diagnosis for industrial robots based on dual-transformer networks, J. Manuf. Syst. 66 (2023) 163–178.
[18] C. Liu, P. Zheng, X. Xu, Digitalisation and servitisation of machine tools in the era of industry 4.0: a review, Int. J. Prod. Res. (2021) 1–33.
[19] Z. Huang, M. Fey, C. Liu, E. Beysel, X. Xu, C. Brecher, Hybrid learning-based digital twin for manufacturing process: Modeling framework and implementation, Robot. Comput.-Integr. Manuf. 82 (2023) 102545.
[20] Q. Liu, Z. Liu, W. Xu, Q. Tang, Z. Zhou, D.T. Pham, Human-robot collaboration in disassembly for sustainable manufacturing, Int. J. Prod. Res. 57 (12) (2019) 4027–4044.
[21] J. Huang, D.T. Pham, Y. Wang, M. Qu, C. Ji, S. Su, W. Xu, Q. Liu, Z. Zhou, A case study in human–robot collaboration in the disassembly of press-fitted components, Proc. Inst. Mech. Eng. B 234 (3) (2020) 654–664.

[22] J. Huang, D.T. Pham, R. Li, M. Qu, Y. Wang, M. Kerin, S. Su, C. Ji, O. Mahomed, R. Khalil, D. Stockton, W. Xu, Q. Liu, Z. Zhou, An experimental human-robot collaborative disassembly cell, Comput. Ind. Eng. 155 (2021) 107189.
[23] M.-L. Lee, S. Behdad, X. Liang, M. Zheng, Task allocation and planning for product disassembly with human–robot collaboration, Robot. Comput.-Integr. Manuf. 76 (2022) 102306.
[24] W. Xu, Q. Tang, J. Liu, Z. Liu, Z. Zhou, D.T. Pham, Disassembly sequence planning using discrete bees algorithm for human-robot collaboration in remanufacturing, Robot. Comput.-Integr. Manuf. 62 (2020) 101860.
[25] S. Parsa, M. Saadat, Human-robot collaboration disassembly planning for end-of-life product disassembly process, Robot. Comput.-Integr. Manuf. 71 (2021) 102170.
[26] I. Elguea-Aguinaco, A. Serrano-Muñoz, D. Chrysostomou, I. Inziarte-Hidalgo, S. Bø gh, N. Arana-Arexolaleiba, Goal-conditioned reinforcement learning within a human-robot disassembly environment, Appl. Sci. 12 (22) (2022) 11610.
[27] M. Chu, W. Chen, Human-robot collaboration disassembly planning for end-of-life power batteries, J. Manuf. Syst. 69 (2023) 271–291.
[28] L. Guo, Z. Zhang, X. Zhang, Human–robot collaborative partial destruction disassembly sequence planning method for end-of-life product driven by multi-failures, Adv. Eng. Inform. 55 (2023) 101821.
[29] J. Liu, Z. Zhou, D.T. Pham, W. Xu, C. Ji, Q. Liu, Robotic disassembly sequence planning using enhanced discrete bees algorithm in remanufacturing, Int. J. Prod. Res. 56 (9) (2018) 3134–3151.
[30] C. Sassanelli, P. Rosa, S. Terzi, Supporting disassembly processes through simulation tools: A systematic literature review with a focus on printed circuit boards, J. Manuf. Syst. 60 (2021) 429–448.
[31] B. Zhou, X. Shen, Y. Lu, X. Li, B. Hua, T. Liu, J. Bao, Semantic-aware event link reasoning over industrial knowledge graph embedding time series data, Int. J. Prod. Res. 61 (12) (2023) 4117–4134.
[32] A. Rega, M. Castrese Di, A. Pasquariello, F. Vitolo, S. Patalano, A. Zanella, A. Lanzotti, Collaborative workplace design: A knowledge-based approach to promote human–robot collaboration and multi-objective layout optimization, Appl. Sci. 11 (24) (2021) 12147.
[33] S. Ong, M. Chang, A. Nee, Product disassembly sequence planning: state-of-the-art, challenges, opportunities and future directions, Int. J. Prod. Res. 59 (11) (2021) 3493–3508.
[34] B. Zhou, J. Bao, J. Li, Y. Lu, T. Liu, Q. Zhang, A novel knowledge graph-based optimization approach for resource allocation in discrete manufacturing workshops, Robot. Comput.-Integr. Manuf. 71 (2021) 102160.
[35] Y. Lu, Q. Shao, C. Singh, X. Xu, X. Ye, Ontology for manufacturing resources in a cloud environment, Int. J. Manuf. Res. 9 (4) (2014) 448–469.
[36] A. Umbrico, A. Orlandini, A. Cesta, An ontology for human-robot collaboration, Procedia CIRP 93 (2020) 1097–1102.
[37] L. Qiao, Y. Qie, Z. Zhu, Y. Zhu, N. Anwer, An ontology-based modelling and reasoning framework for assembly sequence planning, Int. J. Adv. Manuf. Technol. 94 (9) (2018) 4187–4197.
[38] Y. Zhong, C. Jiang, Y. Qin, G. Yang, M. Huang, X. Luo, Automatically generating assembly sequences with an ontology-based approach, Assem. Autom. (2019).
[39] H. Gong, L. Shi, X. Zhai, Y. Du, Z. Zhang, Assembly process case matching based on a multilevel assembly ontology method, Assem. Autom. 42 (1) (2022) 80–98.
[40] B. Zhu, U. Roy, Ontology-based disassembly information system for enhancing disassembly planning and design, Int. J. Adv. Manuf. Technol. 78 (9) (2015) 1595–1608.
[41] B. Zhu, U. Roy, Modeling and validation of a web ontology language based disassembly planning information model, J. Comput. Inf. Sci. Eng. 18 (2) (2018).
[42] G. Foo, S. Kara, M. Pagnucco, An ontology-based method for semi-automatic disassembly of LCD monitors and unexpected product types, Int. J. Autom. Technol. 15 (2) (2021) 168–181.
[43] G. Foo, S. Kara, M. Pagnucco, Artificial learning for part identification in robotic disassembly through automatic rule generation in an ontology, IEEE Trans. Autom. Sci. Eng. (2022).
[44] J.S. Liang, The methodology of knowledge acquisition and modeling for troubleshooting in automotive braking system, Robot. Comput.-Integr. Manuf. 28 (1) (2012) 24–34.
[45] L. Jiang, J. Shi, C. Wang, Multi-ontology fusion and rule development to facilitate automated code compliance checking using BIM and rule-based reasoning, Adv. Eng. Inform. 51 (2022) 101449.
[46] P. Veerakamolmal, S.M. Gupta, A case-based reasoning approach for automating disassembly process planning, J. Intell. Manuf. 13 (2002) 47–60.
[47] F. Giudice, Disassembly depth distribution for ease of service: a rule-based approach, J. Eng. Des. 21 (4) (2010) 375–411.
[48] S. Chen, J. Yi, H. Jiang, X. Zhu, Ontology and CBR based automated decision-making method for the disassembly of mechanical products, Adv. Eng. Inform. 30 (3) (2016) 564–584.
[49] J. Yu, H. Zhang, Z. Jiang, W. Yan, Y. Wang, Q. Zhou, Disassembly task planning for end-of-life automotive traction batteries based on ontology and partial destructive rules, J. Manuf. Syst. 62 (2022) 347–366.
[50] S. Tor, G. Britton, W. Zhang, Y.-M. Deng, Guiding functional design of mechanical products through rule-based causal behavioural reasoning, Int. J. Prod. Res. 40 (3) (2002) 667–682.
[51] C. Zheng, Y. An, Z. Wang, X. Qin, B. Eynard, M. Bricogne, J. Le Duigou, Y. Zhang, Knowledge-based engineering approach for defining robotic manufacturing system architectures, Int. J. Prod. Res. 61 (5) (2023) 1436–1454.
[52] Y. Lu, X. Xu, A semantic web-based framework for service composition in a cloud manufacturing environment, J. Manuf. Syst. 42 (2017) 69–81.
[53] B. Reddy, R. Fields, From past to present: a comprehensive technical review of rule-based expert systems from 1980–2021, in: Proceedings of the 2022 ACM Southeast Conference, 2022, pp. 167–172.
[54] R. Lepratti, Advanced human-machine system for intelligent manufacturing: Some issues in employing ontologies for natural language processing, J. Intell. Manuf. 17 (6) (2006) 653–666.
[55] X. Shen, X. Li, B. Zhou, Y. Jiang, J. Bao, Dynamic knowledge modeling and fusion method for custom apparel production process based on knowledge graph, Adv. Eng. Inform. 55 (2023) 101880.
[56] A. Giovannini, A. Aubry, H. Panetto, M. Dassisti, H. El Haouzi, Ontology-based system for supporting manufacturing sustainability, Annu. Rev. Control 36 (2) (2012) 309–317.
[57] H.S. Pinto, J.P. Martins, Ontologies: How can they be built? Knowl. Inf. Syst. 6 (2004) 441–464.
[58] D.L. McGuinness, F. Van Harmelen, OWL web ontology language overview, W3C Recomm. 10 (10) (2004) 2004.
[59] A. Gómez-Pérez, O. Corcho, Ontology languages for the semantic web, IEEE Intell. Syst. 17 (1) (2002) 54–60.
[60] F. De Fazio, C. Bakker, B. Flipsen, R. Balkenende, The disassembly map: A new method to enhance design for product repairability, J. Clean. Prod. 320 (2021) 128552.
[61] R.K.-J. Lee, H. Zheng, Y. Lu, Human-robot shared assembly taxonomy: A step toward seamless human-robot knowledge transfer, Robot. Comput.-Integr. Manuf. 86 (2024) 102686.
[62] M.-L. Lee, X. Liang, B. Hu, G. Onel, S. Behdad, M. Zheng, A review of prospects and opportunities in disassembly with human–robot collaboration, J. Manuf. Sci. Eng. 146 (2) (2024).
[63] Y. Hu, C. Liu, M. Zhang, Y. Jia, Y. Xu, A conceptual framework of cyber-physical remanufacturing system, in: 2022 8th International Conference on Nanomanufacturing & 4th AET Symposium on ACSM and Digital Manufacturing, Nanoman-AETS, IEEE, 2022, pp. 1–6.