



HAL
open science

Knowledge-based program generation approach for robotic manufacturing systems

Chen Zheng, Jiajian Xing, Zhanxi Wang, Xiansheng Qin, Benoît Eynard, Jing
Li, Jing Bai, Yicha Zhang

► **To cite this version:**

Chen Zheng, Jiajian Xing, Zhanxi Wang, Xiansheng Qin, Benoît Eynard, et al.. Knowledge-based program generation approach for robotic manufacturing systems. *Robotics and Computer-Integrated Manufacturing*, 2022, 73, pp.102242. 10.1016/j.rcim.2021.102242 . hal-04270568

HAL Id: hal-04270568

<https://hal.science/hal-04270568>

Submitted on 22 Jul 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

1 Knowledge-based program generation approach for robotic 2 manufacturing systems

3 Chen Zheng^a, Jiajian Xing^b, Zhanxi Wang^a, Xiansheng Qin^a, Benoît Eynard^b, Jing Li^c, Jing Bai^a,
4 Yicha Zhang^{d,*}

5 ^a*School of Mechanical Engineering, Northwestern Polytechnical University, 127 West Youyi Road,*
6 *Xi'an Shaanxi, 710072, P. R. China*

7 ^b*Shanghai Engineering Center for Microsatellites, 99 Haike Road, Shanghai, 201210, P. R. China*

8 ^c*Department of Mechanical Systems Engineering, Sorbonne Universités, Université de*
9 *Technologie de Compiègne, Roberval Laboratory - UMR CNRS 7337, CS 60319, 60203*
10 *Compiègne Cedex, France*

11 ^d*School of Management, Northwestern Polytechnical University, 127 West Youyi Road, Xi'an*
12 *Shaanxi, 710072, P.R.China*

13 ^e*Mechanical Engineering and Design Department, Université de Bourgogne Franche-Comté,*
14 *Université de Technologie de Belfort-Montbéliard, ICB UMR CNRS 6303, 90010 Belfort Cedex,*
15 *France*

16
17 **Abstract:** In recent decades, robotic manufacturing systems have been considered as effective
18 solutions for providing more productive manufacturing processes, but with less cost and risk.
19 However, the programming for robotic manufacturing systems is a time-consuming task, and
20 hinders the implementation of robotic manufacturing systems in today's industry.

21 This paper proposes a knowledge-based program-generation approach for robotic manufacturing
22 systems. The proposed approach provides effective support for the standardization of the rules and
23 knowledge related to manufacturing programs that have proven successful in previous
24 manufacturing cases; this can not only increase the programming efficiency, but can also improve
25 the manufacturing stability and production quality. First, an ontological knowledge model is
26 developed to provide an explicit semantic description of the relevant concepts for the robotic
27 manufacturing system, basic instruction units for the program, and product models of the
28 workpieces. Second, a rule-based reasoning mechanism is established to infer the implicit
29 relationships between the basic instruction units of the manufacturing program. Finally, based on
30 the semantic descriptions and reasoning mechanism of the proposed knowledge model, the basic
31 instruction units of the manufacturing program are instantiated based on data extracted from the
32 product models and integrated according to the relationships inferred by the reasoning mechanism,
33 thereby generating the robotic manufacturing program.

34 **Keywords:** Knowledge-based engineering; robotic manufacturing system; manufacturing
35 programming; program generation

* yicha.zhang@utbm.fr

1 **1. Introduction**

2 Robotic manufacturing systems have played an important role in speeding up productivity,
3 keeping workers safe, and ensuring higher-quality products over the last few decades. However,
4 there are still some challenges related to the programming task hindering the implementation of
5 robotic manufacturing systems in industry, especially for small and medium-sized enterprises
6 (SMEs). In particular, programming for robotic manufacturing systems requires specific
7 engineering expertise; therefore, it is difficult to implement the robotic manufacturing systems in
8 manufacturing SMEs, owing to the lack of skilled workers in programming [1]. In addition, in a
9 mass customization context where customer needs grow increasingly diverse, different products of
10 very similar types are required, but their respective quantities are sometimes quite small [2]. As a
11 result, programming for robotic manufacturing systems becomes a time-consuming task, as the
12 program needs to be modified even though only small changes are being made for different
13 product types [3]

14 The concept of a robotic manufacturing system was proposed in the 1980s [4], and several
15 subsequent evolution steps have generally been observed. First, programmable logic units, such as
16 programmable logic controllers (PLCs) and other real-time controllers, became symbols of
17 automation in the manufacturing process; programs were generated and sent to the corresponding
18 programmable logic units to control the running status of machining tools [5]. However, such
19 automated manufacturing systems were far from meeting expectations in terms of agility,
20 flexibility, and intelligence [6]. Second, with the large-scale and vigorous development of
21 industrial robots in the 1990s, robotic manufacturing systems began adopting one or more robots
22 to conduct a large number of operations, and various robot programming languages were proposed
23 [7]. Currently, different components (e.g., machine, robot, gantry, and automatic guided vehicle
24 (AGV)) with their own controllers and programming languages have been integrated into robotic
25 manufacturing systems, aiming to further improve the agility, flexibility, and intelligence of the
26 corresponding manufacturing processes [8]. However, owing to the different controlling logics and
27 programming languages adopted by the aforementioned components, coordinating and
28 programming for a robotic manufacturing system remains a challenging task.

29 Focusing on the challenges related to the programming processes for robotic manufacturing
30 systems, the authors herein propose a knowledge-based approach for aiding users in achieving
31 automatic program generation. This paper is organized as follows. Section 2 presents a literature
32 review of current programming approaches. Section 3 introduces the proposed knowledge-based
33 program-generation approach. Section 4 presents an industrial case study based on a robotic
34 welding system, in which the proposed program generation approach is implemented. In Section 5,
35 the main contributions of this study are discussed. Finally, Section 6 concludes the paper.

36 **2. Literature review**

37 **2.1 Decomposition programming approaches**

38 As noted above, different components with their own controllers and programming languages

1 have been integrated into today's robotic manufacturing systems. Owing to the heterogeneity and
2 complexity of the components in robotic manufacturing systems, a robotic manufacturing program
3 is always decomposed into a sequence of basic instruction units; these can be executed by
4 different components, and programmed with specific programming knowledge. Based on the
5 decomposition principle, various programming approaches have been proposed for describing the
6 basic instruction units of a manufacturing program.

7 **Object-oriented programming** is a programming paradigm based on the concept of "objects" in
8 the form of attributes and methods [9]. Various corresponding programming approaches have been
9 proposed, and have shown the possibility of establishing a flexible programming environment
10 based on the object-oriented paradigm for different components of a manufacturing system [10–
11 12]. Each component of a manufacturing system is defined as a class containing a set of attributes
12 and functions. Similar to the object-oriented programming approach, some researchers have
13 utilized the term "agent" to describe the aforementioned basic instruction units, and have proposed
14 the so-called **agent-based programming approach**. An agent is defined as a composition of
15 states; it comprises social, environmental, and personal knowledge, a perception function, and
16 behavioral rules [13]. The agent-based programming approach is therefore a distributed
17 programming method that decomposes a manufacturing process into different tasks performed by
18 agents, in which each agent is described with parameters, local variables, and a list of commands
19 [14,15].

20 Some researchers believe that an excessively fine decomposition hierarchy may lead to a vast
21 number of instruction units for manufacturing programs, thereby resulting in time-consuming
22 work for a large number of tiny codes [16]. To reduce the amount of data and improve the
23 reusability of the basic instruction units, AlGeddawy proposed a **modular programming**
24 **approach** using the Design Structure Matrix (DSM) to help users determine the most suitable
25 program decomposition level [17]. [Based on the modular programming principle, the function](#)
26 [block](#), defined as the encapsulation of a machine process data and function, was initially proposed
27 [by the International Electrotechnical Commission for industrial process measurement and control](#)
28 [systems \[18\]. To facilitate the programming task for the computer-numerically controlled \(CNC\)](#)
29 [machine tools of a manufacturing system, Xu et al. proposed that CNC machine tools could be](#)
30 [decomposed into modularized function blocks; these could then be integrated to fulfill specific](#)
31 [manufacturing tasks\[19\]. In addition to CNC machines, Wang et al. considered industrial robots,](#)
32 [and utilized a function block for the machining and assembly control of robotic manufacturing](#)
33 [systems \[20\]. By taking advantage of the autonomy and cooperation of function blocks, Yao et al.](#)
34 [considered human activities during the manufacturing process, and studied the human-robot](#)
35 [collaboration in robotic manufacturing systems \[21\].](#)

36 The above-mentioned programming approaches based on the decomposition principle have
37 facilitated the programming task for robotic manufacturing systems to a great extent. However,
38 users still need to call and parameterize the predefined instruction units using certain commands
39 [22]; as a result, this type of programming approach can hardly support automatic program
40 generation for robotic manufacturing systems.

41

1

2 **2.2 Automatic programming approaches**

3 Currently, two different programming modes have been developed and are widely used to achieve
4 automatic programming for industrial robots in robotic manufacturing systems, i.e., online and
5 offline modes [23]. **Online programming** is usually realized by using a teaching programming
6 method [24]. The principles of teaching programming methods can be summarized as follows.
7 When applying the online programming approach, the robot is manually controlled by an operator
8 to achieve a series of required motions, and the information and data of which are recorded and
9 edited so that the program can be generated automatically. The online mode not only provides
10 users with a natural way to program an industrial robot which is easy to learn and requires no
11 programming skill or education, but also makes programming safer, as the robot speed and motion
12 can be limited and coordinated manually by considering the positions of the other pieces of
13 equipment in the robotic manufacturing system. However, users must sometimes spend a
14 significant amount of time applying such teaching programming methods when the product is
15 complex in nature and/or many points are involved. To improve the efficiency of the online
16 programming approach, Kohrt et al. presented an online programming support system based on
17 the Voronoi roadmap and elastic net; the system transformed user interactions into simplified tasks
18 applicable to industrial robot programming, so that the robot programming time could be reduced
19 [25]. Zhang et al. proposed an online programming method for industrial robots by combining
20 vision, force, and position sensors. By using a force and visual servoing strategy, the industrial
21 robot was controlled to follow desired trajectories [26]. Hein et al. developed a modular online
22 programming environment in which the user could conduct tele-operations, and his/her motion
23 could be quickly interpreted and executed as a robot motion [27]. Neto et al. attached
24 accelerometers to a user's body to capture arm movements and teach a robot by performing
25 gestures [28]. Although online programming provides users with a simple approach to generating
26 a program for industrial robots, a completely new program generally should be written if a new
27 workpiece is processed. This is time-consuming and may lead to a critical loss in production
28 efficiency, especially for manufacturing SMEs where diverse workpieces are produced, but their
29 batch sizes are sometimes quite small.

30 Off-line programming is an automatic programming generation technique that avoids adopting a
31 real robot [29], and can be generally divided into two categories: computer-aided design
32 (CAD)-based offline programming approaches, and vision-based offline programming approaches
33 [30]. **CAD-based offline programming** helps users generate robot programs based on CAD
34 models of workpieces. Kim et al. proposed a PC-based offline programming approach, and
35 developed a CAD interface to generate a program for welding robots in shipbuilding [31].
36 Zivanovic et al. translated and transferred a CAD/computer-aided manufacturing model under the
37 STEP-NC standard to generate robot programming for machining operations [32]. Ding et al. used
38 CAD models as inputs to generate deposition paths and manufacturing parameters for industrial
39 robots, aiming to achieve an automated arc-welding-based additive manufacturing process [33].
40 Lu and Lin integrated online calibration into an offline programming approach to calibrate relative
41 location errors [34]. Current industrial robotics manufacturers (e.g., ABB and KUKA) and
42 industrial software companies (e.g., Dassault Systèmes) also provide users with their own offline

1 programming software (ABB RobotStudio¹ and KUKA.Sim²) or platforms (DELMIA Robotics³)
2 to increase the efficiency of offline programming [35]. However, owing to the syntax differences
3 in the various robot programming languages, the output code from commercial offline
4 programming software generally needs to be post-processed into a target robot language [36].

5 **Vision-based offline programming approaches** adopt vision sensors to obtain the geometric
6 features of a product, determine robot motions, and generate programs. Pei and Chen applied
7 machine vision technology to a robotic welding production line to aid robots in identifying
8 welding components and selecting corresponding programs [37]. Jia et al. proposed the use of
9 monocular-structured light vision to detect welding seams and achieve automatic trajectory
10 planning for a welding robot [38]. Yang et al. developed a vision sensor for reconstructing a 3D
11 model of workpieces and determining a robot path [39]. In general, compared to online
12 programming approaches, offline programming approaches are intended to be more flexible and
13 productive when facing complex products. However, offline programming often requires special
14 techniques or commercial offline programming software. For example, skilled workers with
15 specific engineering expertise are often required for program optimization and debugging;
16 moreover, commercial offline programming software is always highly expensive, increasing the
17 financial burden on manufacturing companies.

18 **2.3 Knowledge-based industrial robotics**

19 Robotic manufacturing systems incorporate mechanical engineering, electrical engineering,
20 software engineering, and control engineering. The hardware and software are coupled to each
21 other in robotic manufacturing systems; generally, software designers can only adjust the software
22 parts, without touching the hardware parts. Notably, the designers of robotic manufacturing
23 systems are not hesitant to investigating the hardware and trying to fix it when something goes
24 wrong. However, when facing programming problems, they are not as good as professional
25 software developers. At any rate, designers are always less interested in the low-level components,
26 such as those for communication, coordination, and synchronization, which comprise the entire
27 robotic manufacturing system. On the contrary, they pay more attention to the high-level
28 capabilities offered by the robotic manufacturing system, such as navigation, manipulation, and
29 other specific tasks. Against this background, a robot operating system (ROS) has been developed
30 to aid designers in programming a robotic manufacturing system, based on simply knowing the
31 ROS application program interface (API) [40]. The ROS is not a traditional operating system
32 responsible for process management and scheduling; instead, it simplifies the creation of robot
33 behaviors, frees the operators from the burden of managing the communications or coordination
34 between components, and promotes the decomposition of their functionalities [41]. Owing to the
35 ROS, designers with limited knowledge of programming are able to control a robot; as a result,
36 ROSs have been widely used in education settings to facilitate robot design and implementation

¹ <https://new.abb.com/products/robotics/robotstudio>

² https://www.kuka.com/en-us/products/robotics-systems/software/simulation-planning-optimization/kuka_sim

³ <https://www.3ds.com/events/single-eseminar/delmia-robotic-simulation-and-offline-programming-solutions/>

1 [42]. However, a series of experiments were conducted by Mayoral-Vilches et al. indicated that
2 ROSs were not yet favorable for secure use in industry today [43].

3 In addition to a ROS and relevant APIs, the availability of a knowledge-based engineering (KBE)
4 approach can also simplify the programming process to meet the programming requirements of the
5 designers of the robotic manufacturing system. Currently, semantic technology is widely used to
6 represent and transmit knowledge in a formal, machine-readable structure; therefore, it is
7 considered the main pillar of the KBE approach. The term semantics refers to the process of
8 communicating sufficient meaning to result in an action [44]. Therefore, with the support provided
9 by the explicit semantic relationships in a knowledge model, different systems, machines, and
10 users can easily understand the meaning of the knowledge [45]; such models have been used in
11 many application areas, such as assembly sequence planning [46], conceptual designs of complex
12 systems [47], and material selection for mechanical components [48]. As one of the building
13 blocks of semantic technology, an ontology is generally adopted to provide a formal knowledge
14 description. In the domain of robotics, different ontologies have been proposed for supporting
15 robot autonomy [49]. The ROSETTA project funded by the European Union developed an
16 ontology focusing on robotics devices and skills [50]. Based on this ontology, Stenmark and
17 Malec presented a knowledge-based approach to instructing manipulation tasks for industrial
18 robots [51], and Patel et al. adopted the shared knowledge between engineering systems and
19 robots to adapt to new manufacturing tasks when a production line was changed to produce new
20 products [52]. The Institute of Electrical and Electronics Engineers Robotics and Automation
21 Society working group introduced a core ontology named “Ontologies for Robotics and
22 Automation” (ORA) to structure the terms commonly used in the domain of robotics and
23 automation [53]. Subsequently, various extensions of the ORA have been proposed for specifying
24 terms across different subdomains of robotics. Fiorini et al. complemented the ORA with terms
25 relating to the position, orientation, and pose of a robot [54]. Jorge et al. further extended the ORA
26 and defined spatial terms for representing both heterogeneous robots and human-robot interactions
27 [55].

28 The operators of a robotic manufacturing system are always considered as the most suitable for
29 programming certain manufacturing processes, as they are more familiar with the required
30 manufacturing processes. Although the above-mentioned KBE approaches have already greatly
31 reduced the programming complexity and enabled non-experts to program robotic manufacturing
32 systems, owing to a lack of necessary expertise in robotics, programming for robotic
33 manufacturing systems remains a mentally intensive and challenging task for operators. In other
34 words, although knowledge-based industrial robotics offers enormous potential for vastly
35 improving robot autonomy, to the authors’ knowledge, the current KBE approaches cannot fully
36 support automatic program generation for robotic manufacturing systems.

37

38

39

1

2 2.4 Summary of literature review

3 After analyzing the current programming approaches for robotic manufacturing systems, the
4 authors present the main limitations of each programming approach in Table 1.

5 Table 1 Main limitations of current programming approaches

Programming approaches		Limitations	
Decomposition programming approaches	Object-oriented programming [10–12]	Decomposition programming approaches can hardly support the automatic program generation for robotic manufacturing systems.	
	Agent-based programming [14, 15]		
	Function block based programming [18–21]		
Automatic programming approaches	On-line programming [25–28]	It is time-consuming and may lead to critical loss in production efficiency if diverse workpieces with small batch sizes are produced. 1. Skilled workers with specific engineering expertise are required. 2. Commercial off-line programming software is always highly expensive.	
	Off-line programming		Computer-aided design (CAD)-based off-line programming [32–35]
			Vision-based off-line programming approach [37–39]
Knowledge-based industrial robotics	Robot operating system (ROS)-based programming [40–42]	The secure use of ROS in industry cannot be guaranteed.	
	Current knowledge-based engineering approaches [50–55]	Current knowledge-based engineering approaches cannot fully support automatic program generation for robotic manufacturing systems.	

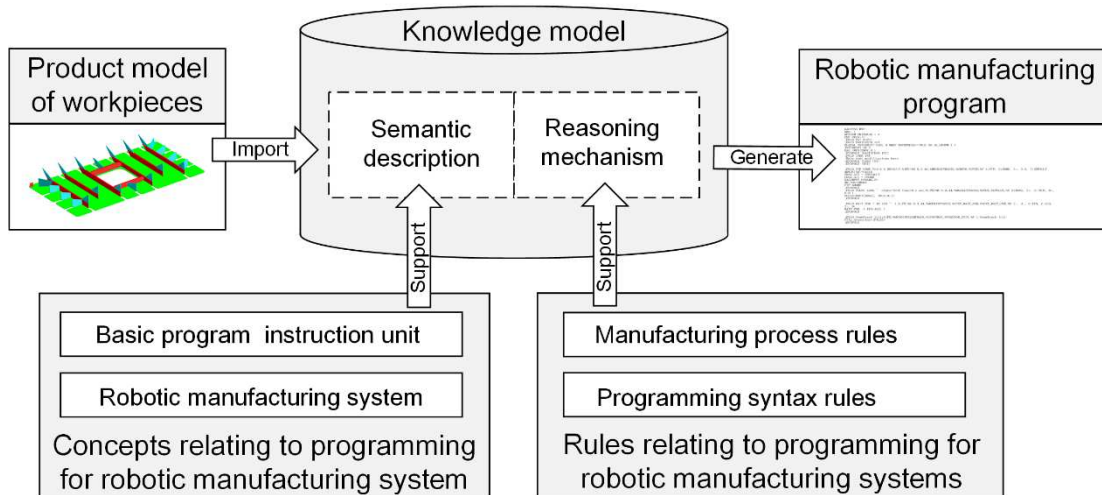
6 As mentioned above, researchers have proposed different programming approaches based on the
7 decomposition principle, thereby facilitating the programming task for robotic manufacturing
8 systems to a great extent. However, users generally call and parameterize predefined instruction
9 units using certain commands. As a result, this type of programming approach cannot achieve
10 automatic programming generation. Currently, online programming and offline programming
11 approaches have been widely used to aid users in realizing automatic programming for industrial
12 robots. As mentioned above, in the online programming approaches, the program is generated by
13 manually controlling the robot to achieve a series of required motions; it is therefore
14 time-consuming, and may lead to a critical loss in production efficiency if diverse workpieces with
15 small batch sizes are produced. Although offline programming provides a more flexible and
16 productive approach when facing complex products, skilled workers with specific engineering
17 expertise are required to extract the necessary information from the CAD models or image
18 features of the workpieces, optimize the program, and debug the codes. In addition, a robotic
19 manufacturing system usually requires a set of different commercial offline programming software
20 tools, thereby incurring considerable costs for SMEs. Moreover, most of the current online and
21 offline approaches are only for programming industrial robots; nevertheless, an industrial robot
22 alone is not a real robotic manufacturing system, and sometimes is only a part of a robotic
23 manufacturing system. To help designers with limited programming knowledge in controlling a
24 robot, ROSs have been proposed and widely used in educational settings to facilitate robot design
25 and implementation; however, the secure use of ROSs in industry cannot be guaranteed. As for the
26 KBE approaches, although current KBE approaches offer enormous potential for vastly improving

1 robot autonomy, none of them can fully support automatic program generation for robotic
2 manufacturing systems.

3 Therefore, a programming approach that can fully support automatic program generation for
4 coordinating an entire robotic manufacturing system should be further investigated.

5 3. Knowledge-based program generation approach

6 As presented in Section 2, based on the decomposition principle, users can call, parameterize, and
7 integrate predefined instruction units to generate a program for robotic manufacturing systems.
8 Although this type of programming approach does not help users achieve automatic program
9 generation, it still greatly facilitates the programming tasks for robotic manufacturing systems.
10 Moreover, current offline programming approaches, such as CAD-based programming approaches,
11 already provide users with automatic programming, but skilled workers with specific engineering
12 expertise are required to extract the necessary information from the CAD models of the
13 workpieces. Considering the advantages and limitations of the two different categories of
14 programming approaches, the authors herein propose a knowledge-based approach by integrating
15 the decomposition principle into an offline programming approach with the use of semantic
16 technology, so as to achieve automatic program generation for robotic manufacturing systems.
17 With the support provided by the explicit semantic descriptions in the knowledge model, the
18 concepts in the relevant domains (e.g., domains of workpieces, robotic manufacturing processes,
19 robotic manufacturing systems, and robotic manufacturing program) can be easily understood by
20 users and processed by computers. In addition, the rules are represented in a formal semantic
21 description and proposed according to the intra- and inter-domain relationships; based on these,
22 a reasoning mechanism is established to infer the sequence of basic instruction units, thereby
23 generating the required robotic manufacturing program (Fig. 1).



24

25

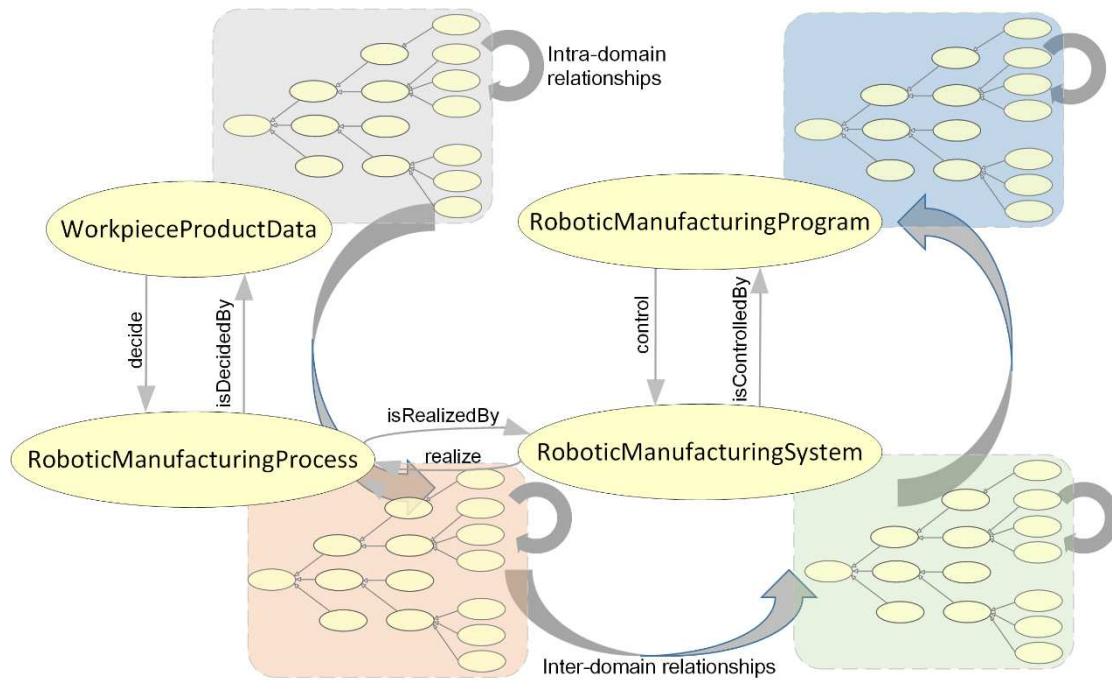
Fig. 1 Framework of knowledge-based program generation approach

26

3.1 Semantic description for concepts of robotic manufacturing systems

1 The term “semantics” refers to the process of communicating sufficient meaning to result in an
2 action [44]. Owing to the lack of semantic descriptions and relationships for the concepts relating
3 to robotic manufacturing systems, basic program instruction units, and product models of
4 workpieces, skilled workers with specific engineering expertise are needed to utilize the relevant
5 concepts when implementing the current offline programming approaches. Semantic web
6 technology has been proposed and widely used to represent and transmit information in a formal,
7 machine-readable structure [56]. With the support of semantic web technology, both users and
8 computers can easily understand and process the above-mentioned concepts. As one of the
9 building blocks of semantic web technology, ontology is adopted in this study to provide a formal
10 semantic description of the concepts are closely related to programming for robotic manufacturing
11 systems. In particular, the authors adopt ⁴ the Web Ontology Language (OWL) developed by the
12 World Wide Web Consortium to provide a semantic description of the relevant concepts.
13 According to the framework of the knowledge-based program generation approach, the four root
14 concepts (i.e., robotic manufacturing system, processes, program, and product data of workpieces)
15 and their semantic relationships in the proposed knowledge model are presented in Fig. 2. The
16 manufacturing process is decided by comprehensively considering the workpiece geometry,
17 manufacturing sequence, and technical requirements stored in the product data of the workpieces.
18 The robotic manufacturing system realizes the required manufacturing processes, and the program
19 commands the robotic manufacturing system to perform certain manufacturing processes.
20 Accordingly, the relationship between the *WorkpieceProductData* and
21 *RoboticManufacturingProcess* classes can be represented through the object properties *decide* and
22 *isDecidedBy*. The object properties between *decide* and *isDecidedBy* are defined as inverse
23 properties by using the property description *owl: inverseOf*. The same principle is adopted to
24 describe the relationship between the *RoboticManufacturingProcess* and
25 *RoboticManufacturingSystem* classes, or the *RoboticManufacturingSystem* and
26 *RoboticManufacturingProgram* classes. The details relating to the four root concepts are presented
27 in the following subsections.

⁴<https://www.w3.org/OWL/>

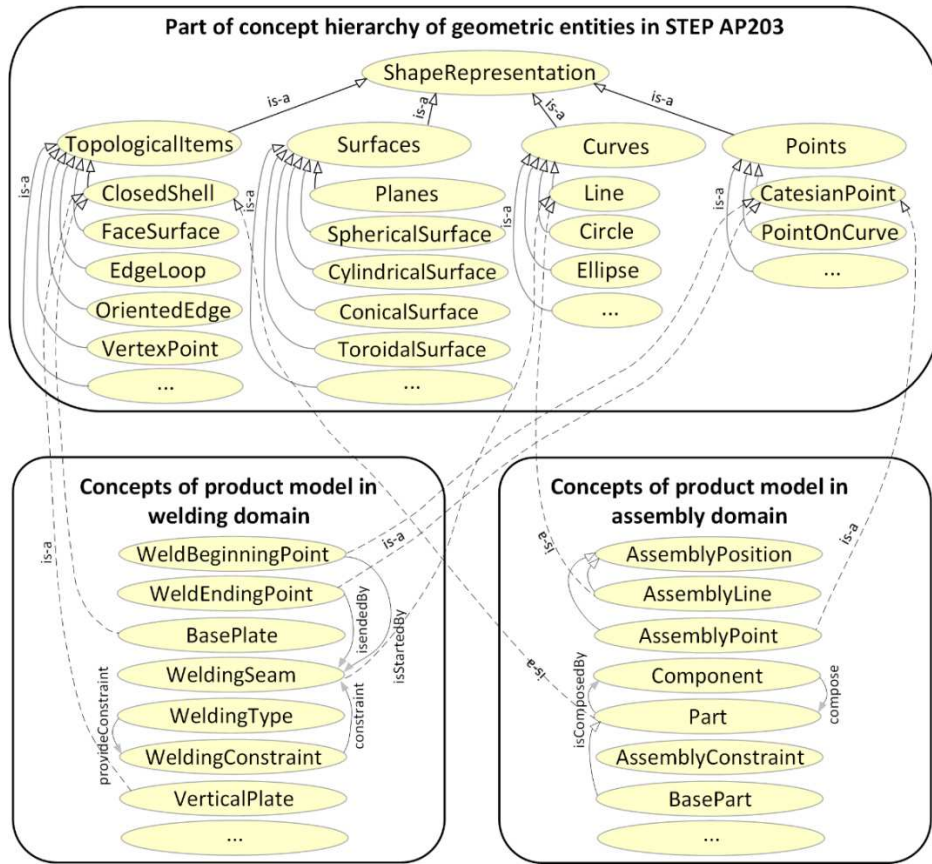


1

2 Fig. 2 Semantic relationships among robotic manufacturing system, process, program, and
3 workpieces

4 **3.2 Part of workpiece product data in ontological knowledge model**

5 Information relating to workpiece geometry, manufacturing sequences, and specific technical
6 requirements can be found in the product models of the workpieces. Various product models, such
7 as the STandard for the Exchange of Product model (STEP) [57], core product model (CPM) [58],
8 and product-process-organization model (PPO) [59], have been proposed for product development,
9 aiming to provide a common information expression form for all engineers during the product
10 development process. However, the meanings behind the expressions are not provided, owing to
11 the lack of semantic description mechanisms in the current product models. OntoSTEP was
12 proposed as a geometry-enhanced ontological model for enriching the meaning behind the
13 geometric data in a STEP file [60]. It can be used as a fundamental ontological model for
14 describing the concepts relating to the geometric information and shape representations in the
15 product models of workpieces, so as to minimize the semantic uncertainty among heterogeneous
16 manufacturing applications [61]. Currently, different robotic manufacturing systems are being
17 developed to conduct various manufacturing tasks. As a result, part of the workpiece product data
18 in the proposed ontological knowledge model can be achieved by integrating the concepts and
19 semantic relationships into the current geometry-enhanced ontological model (e.g., OntoSTEP).



1

2

Fig. 3 Representation of concepts relating to product data of workpieces in different manufacturing domains

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

Fig. 3 presents the concepts relating to the product data of workpieces in different manufacturing domains (i.e., robotic assembling and welding domains). Some of the fundamental concepts relating to the shape representations and their relationships defined in STEP AP203 are illustrated in the upper part of Fig. 3. For example, the *Point* class represents a specific position in space, and its subtypes represent various ways to define this position. The *CartesianPoint* class adopts a list of coordinates to specify the position, whereas the *PointOnCurve* class defines the point's location by referring to a specific curve (instance of the *Curve* class) and a parameter value of the point on the curve. The relationship between the *Point* class and its subtypes is described by a semantic link "is-a" in the ontology. To satisfy the requirements for various manufacturing tasks, the concepts of specific manufacturing domains should be integrated into the fundamental geometric ontology. The concepts of two specific robotic manufacturing domains, i.e., the robotic assembly and welding domains, are presented as an example in the lower part of Fig. 3. The explicit relationships between the concepts of fundamental geometric entities and those of specific manufacturing domains can directly be described by the object properties, and the Semantic Web Rule Language (SWRL) can be adopted to define the implicit relationships. Table 2 presents some examples of implicit intra-domain relationships between concepts in the domain of robotic welding systems as described in the SWRL.

Table 2 Examples of implicit intra-domain relationships description for welding application

Implicit relationship	Description in SWRL
-----------------------	---------------------

description	
The horizontal welding seam shall be the contacted edge of the vertical plate and base plate.	$BasePlate(?bp) \wedge VerticalPlate(?vp) \wedge contact(?bp, ?vp) \wedge Line(?l) \wedge Edge(?e) \wedge isComposedBy(?l, ?bp) \wedge isComposedBy(?e, ?vp) \wedge isComposedBy(?l, ?e) \rightarrow HorizontalWeldingSeam(?l)$
The vertical welding seam shall be the contacted edge of the two vertical plates.	$VerticalPlate(?vp1) \wedge VerticalPlate(?vp2) \wedge contact(?vp1, ?vp2) \wedge Line(?l) \wedge Edge(?e) \wedge isComposedBy(?l, ?vp1) \wedge isComposedBy(?e, ?vp2) \wedge isComposedBy(?l, ?e) \rightarrow VerticalWeldingSeam(?l)$
The beginning point of welding process for vertical welding seam shall be the vertex of welding seam whose value of Z-coordinate is smaller.	$VerticalWeldingSeam(?l) \wedge EdgeStart(?p1) \wedge isComposedBy(?p1, ?l) \wedge EdgeEnd(?p2) \wedge isComposedBy(?p2, ?l) \wedge ZCoordinate(?z1) \wedge isComposedBy(?z1, ?p1) \wedge ZCoordinate(?z2) \wedge isComposedBy(?z2, ?p2) \wedge isGreaterThan(?z2, ?z1) \rightarrow WeldBeginningPoint(?p1)$
The beginning point of welding process for horizontal welding seam shall be the vertex of welding seam whose value of X-coordinate is smaller if the welding seam is not parallel to X-axis.	$HorizontalWeldingSeam(?l) \wedge EdgeStart(?p1) \wedge isComposedBy(?p1, ?l) \wedge EdgeEnd(?p2) \wedge isComposedBy(?p2, ?l) \wedge ZCoordinate(?z1) \wedge isComposedBy(?z1, ?p1) \wedge ZCoordinate(?z2) \wedge isComposedBy(?z2, ?p2) \wedge isEqualTo(?z2, ?z1) \wedge XCoordinate(?x1) \wedge isComposedBy(?x1, ?p2) \wedge XCoordinate(?x2) \wedge isComposedBy(?x2, ?p2) \wedge \neg isEqualTo(?x1, ?x2) \wedge YCoordinate(?y1) \wedge isComposedBy(?y1, ?p1) \wedge YCoordinate(?y2) \wedge isComposedBy(?y2, ?p2) \wedge isGreaterThan(?y2, ?y1) \rightarrow WeldBeginningPoint(?p1)$
The beginning point of welding process for horizontal welding seam shall be the vertex of welding seam whose value of Y-coordinate is smaller if the welding seam is parallel to X-axis.	$HorizontalWeldingSeam(?l) \wedge EdgeStart(?p1) \wedge isComposedBy(?p1, ?l) \wedge EdgeEnd(?p2) \wedge isComposedBy(?p2, ?l) \wedge ZCoordinate(?z1) \wedge isComposedBy(?z1, ?p1) \wedge ZCoordinate(?z2) \wedge isComposedBy(?z2, ?p2) \wedge isEqualTo(?z2, ?z1) \wedge XCoordinate(?x1) \wedge isComposedBy(?x1, ?p2) \wedge XCoordinate(?x2) \wedge isComposedBy(?x2, ?p2) \wedge isEqualTo(?x1, ?x2) \wedge YCoordinate(?y1) \wedge isComposedBy(?y1, ?p1) \wedge YCoordinate(?y2) \wedge isComposedBy(?y2, ?p2) \wedge isGreaterThan(?y2, ?y1) \rightarrow WeldBeginningPoint(?p1)$

1

2 3.3 Part of robotic manufacturing process in ontological knowledge model

3 The robotic manufacturing process is one of the most important factors, as it determines the
4 manufacturing sequence by which the final product can be realized. Currently, process modeling
5 approaches are proposed based on two modeling principles [62]: state-based and action-based
6 process modeling principles. Fig. 4 presents the two types of process modeling principles. A
7 state-based process modeling principle focuses on the transition from one state to another. As a
8 result, approaches based on the state-based process modeling principles can be used to describe all
9 possible states that one particular component (or even the entire robotic manufacturing system)
10 can reach. An action-based process modeling principle focuses on the control flow from one
11 action to another, and is much closer to the logic and syntax of current robotic manufacturing
12 programming languages. Therefore, to provide effective support for programming robotic
13 manufacturing systems, the authors define part of the robotic manufacturing process of the
14 ontological model herein based on an action-based process modeling principle.

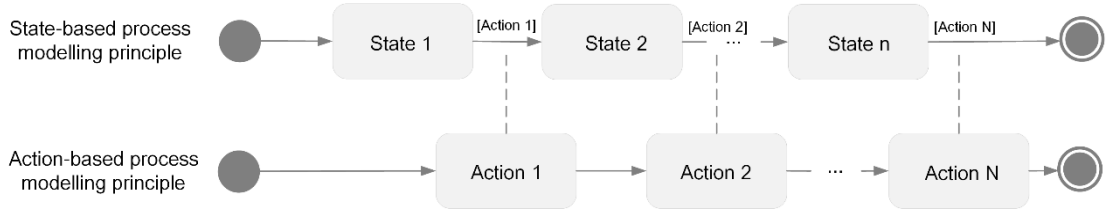


Fig. 4 State-based and action-based process modelling principles

Fig. 5 presents the concept hierarchy relating to the part of robotic manufacturing process described by the ontology. Based on the action-based process modelling principle, the manufacturing process is considered as a sequence of different manufacturing activities. As a result, the activities conducted by different modules and components of the robotic manufacturing system, e.g., *RobotActivity*, *GantryActivity*, and *AGVActivity*, are considered as the basic classes of the concept hierarchy. Five types of relationships, i.e., *is-a*, *isComposedof*, *isBefore*, *isAfter*, and *isIncompatibleWith* are defined as object properties to provide additional details regarding the relationships between the proposed concepts. Table 3 illustrates the object properties defined for the robotic manufacturing process. Similar to the implicit relationships presented in Table 2 of Section 3.2, the implicit relationships between concepts relating to the robotic welding process can be represented based on the proposed object properties.

Table 3 Object properties defined for robotic manufacturing process

Object property	Description
<i>is-a</i> (?a, ?b)	The activity <i>a</i> is a type of activity <i>b</i> .
<i>isComposedBy</i> (?a, ?b)	The activity <i>b</i> is the sub-process of activity <i>a</i> .
<i>isBefore</i> (?a, ?b)	The activity <i>a</i> should be conducted before activity <i>b</i> .
<i>isAfter</i> (?a, ?b)	The activity <i>a</i> should be conducted after activity <i>b</i> .
<i>isIncompatibleWith</i> (?a, ?b)	If the activity <i>a</i> is conducted, the activity <i>b</i> should not be conducted.

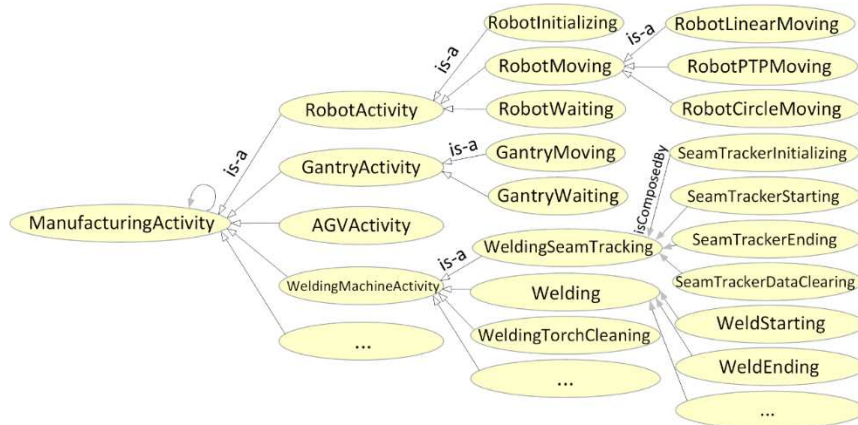


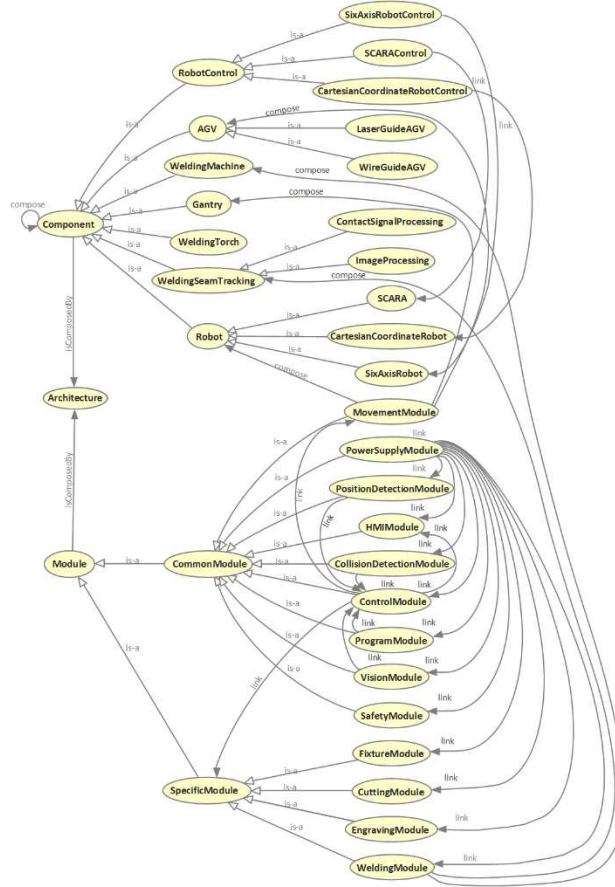
Fig. 5 Representation of concepts relating to robotic manufacturing process

3.4 Part of robotic manufacturing system in ontological knowledge model

The robotic manufacturing programs vary for different manufacturing tasks. Moreover, even though the same manufacturing task may be required by customers (e.g., a robotic welding task),

1 the manufacturing programs should be modified according to different architectures of robotic
2 manufacturing systems (e.g., single-robot and two-robot welding systems). Therefore, in addition
3 to the product data of the workpieces, the concepts relating to the architecture also play a
4 significant role in programming robotic manufacturing systems. Fig. 6 presents the concept
5 hierarchy for robotic manufacturing systems described by the ontology. A module of a robotic
6 manufacturing system refers to the encapsulation of a functional realization, and is a composition
7 of components for satisfying the requirements of users under a set of constraints. Therefore, the
8 *architecture*, *module*, and *component* classes are defined as three basic concepts. The *module* class
9 is divided into *CommonModule* and *SpecificModule* classes. The *CommonModule* class represents
10 the modules demanded by every robotic manufacturing system, such as the *controller module*,
11 *power supply module*, and *movement module* classes. The *SpecificModule* class refers to types of
12 modules adapted to complete specific manufacturing tasks, such as the *WeldingModule* class
13 defined for robotic welding systems.

14 Similar to the part of the workpiece product data presented in Section 3.2, the authors adopt object
15 properties to define the semantic relationships between the different classes of the robotic
16 manufacturing system in the ontological knowledge model, and to help designers draw meaning
17 from these classes. For example, one module is composed of several components, and one
18 component can be further decomposed into several sub-components. Therefore, the
19 “*isComposedBy*” relationship is defined to present the relationships between a module and
20 component, or between a component and sub-component. The “*power*” relationship is used to
21 indicate that the power supply module provides electric power to other modules, whereas the
22 “*control*” relationship illustrates that the control module of one robotic manufacturing system
23 commands and regulates the behaviors of other modules or components.



1

2

Fig. 6 Representation of concepts relating to robotic manufacturing systems

3

3.5 Part of robotic manufacturing program in ontological knowledge model

4

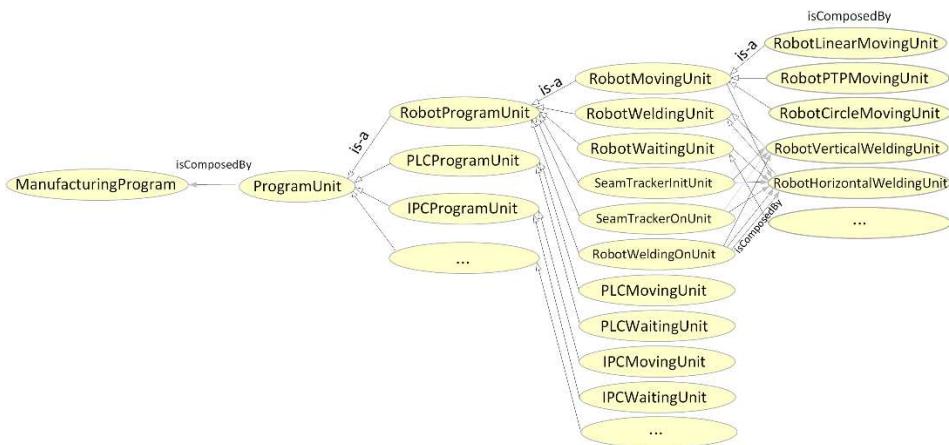
According to the analysis results regarding the current robotic manufacturing programming approaches presented in Section 2, a program for the different control parts of a robotic manufacturing system, for example, a robot controller, PLCs, and other real-time controllers, can be decomposed into a sequence of basic instruction units. Based on the decomposition principle, a concept hierarchy related to the robotic manufacturing program is shown in Fig. 7.

5

6

7

8



9

10

Fig. 7 Representation of concepts relating to robotic manufacturing program

1 The relationships *isAfter* and *isBefore* are defined as object properties for representing the
2 sequences of the different program units. Sequences relating to certain program units can be
3 determined according to basic programming logic. For example, the welding seam tracker should
4 be initialized before starting the seam tracking task, so the *WeldingSeamTrackerInitializingUnit*
5 should be executed before *the WeldingSeamTrackingUnit*. However, some sequences should be
6 determined by comprehensively considering the interactions among robotic manufacturing
7 systems, robotic manufacturing processes, and the product data of workpieces. Hence,
8 inter-domain relationships are proposed and adopted to infer the sequences of basic instruction
9 units in the required robotic manufacturing program(s). Table 4 presents some examples of the
10 implicit inter-domain relationships described by the SWRL.

11 With the support provided by the explicit semantic descriptions for the product models of
12 workpieces, robotic manufacturing processes, robotic manufacturing systems, and robotic
13 manufacturing programs, the concepts can be easily understood by users and processed by
14 computers. A rule-based reasoning mechanism can then be established based on the intra- and
15 inter-domain relationships to infer the sequences of the basic instruction units of the required
16 robotic manufacturing program. Based on the proposed semantic descriptions and reasoning
17 mechanism, the basic instruction units of the manufacturing program can be instantiated by the
18 data extracted from the relevant domains and integrated according to a suitable sequence which
19 can be queried through a semantic query-enhanced web rule language (SQWRL), thereby
20 generating the robotic manufacturing program.

21 In this section, a knowledge-based program generation approach for robotic manufacturing is
22 illustrated. To demonstrate its effectiveness, the following section discusses an application of the
23 proposed approach to a design case of a robotic welding system.

24 Table 4 Examples of implicit inter-domain relationship descriptions

Implicit relationship description	Description in SWRL
The horizontal welding seam whose length exceeds the reach of industrial robot shall be welded by the gantry controlled by the programmable logic controller (PLC).	<i>ProgramUnit(?pu) ∧ RoboticWeldingSystem(?rws) ∧ WeldingProgram(?wpm) ∧ WeldingProcess(?wps) ∧ isComposedBy(?pu, ?wpm) ∧ WeldingSeam(?ws) ∧ isControlledBy(?rws, ?wpm) ∧ isRealizedBy(?wps, ?rws) ∧ isDecidedBy(?wps, ?ws) ∧ Length(?l) ∧ hasLength(?ws, ?l) ∧ Robot(?r) ∧ isComposedBy(?r, ?rws) ∧ hasRange(?r, ?rr) ∧ RobotRange(?rr) ∧ isGraterThan(?l, ?rr) → PLCWeldingUnit(?pu)</i>
If the ending point of a horizontal welding seam is noted as a starting point of all-round welding, the all-round fillet welding unit shall be executed after the horizontal welding process.	<i>ProgramUnit(?pu) ∧ HorizontalWeldingUnit(?hwu) ∧ isAfter(?pu, ?hwu) ∧ HorizontalWeldingSeam(?hws) ∧ WeldEndingPoint(?wep) ∧ isComposedBy(?wep, ?hws) ∧ AllRoundWeldStartingPoint(?arwsp) ∧ isEqualTo(?arwsp, ?wep) ∧ RoboticWeldingSystem(?rws) ∧ WeldingProgram(?wpm) ∧ WeldingProcess(?wps) ∧ isComposedBy(?pu, ?wpm) ∧ isControlledBy(?rws, ?wpm) ∧ isRealizedBy(?wps, ?rws) ∧ isDecidedBy(?wps, ?ws) → AllRoundFilletWeldingUnit(?pu)</i>
The vertical welding seam should be welded after the	<i>VerticalWeldingUnit(?vwu) ∧ HorizontalWeldingUnit(?hwu) ∧ WeldingProgram(?wpm) ∧ isComposedBy(?vwu, ?wpm) ∧ isComposedBy(?hwu, ?wpm) ∧ RoboticWeldingSystem(?rws) ∧ Welding</i>

welding tasks for horizontal welding seams are finished.	$Process(?wps) \wedge isComposedBy(?pu, ?wpm) \wedge isControlledBy(?rws, ?wpm) \wedge isRealizedBy(?wps, ?rws) \wedge isDecidedBy(?wps, ?ws) \rightarrow isAfter(?vwu, ?hwu)$
The welding program unit for the horizontal welding seam whose X-coordinate of the beginning point is smaller shall be executed before the program unit for the welding seam which possesses a larger X-coordinate of the beginning point.	$HorizontalWeldingSeam(?hws1) \wedge HorizontalWeldingSeam(?hws2) \wedge WeldBeginningPoint(?wbp1) \wedge WeldBeginningPoint(?wbp2) \wedge isComposedBy(?wbp1, ?hws1) \wedge isComposedBy(?wbp2, ?hws2) \wedge HorizontalWeldingProcess(?hwps1) \wedge HorizontalWeldingProcess(?hwps2) \wedge isDecidedBy(?hwps1, ?hws1) \wedge isDecidedBy(?hwps2, ?hws2) \wedge HorizontalWeldingProgramUnit(?hwpu1) \wedge HorizontalWeldingProgramUnit(?hwpu2) \wedge RoboticWeldingSystem(?rws) \wedge WeldingProgram(?wpm) \wedge WeldingProcess(?wps) \wedge isComposedBy(?hwpu1, ?wpm) \wedge isComposedBy(?hwpu2, ?wpm) \wedge isComposedBy(?hwps1, ?wps) \wedge isComposedBy(?hwps2, ?wps) \wedge isControlledBy(?rws, ?wpm) \wedge isRealizedBy(?wps, ?rws) \wedge isDecidedBy(?wps, ?ws) \wedge XCoordinate(?x1) \wedge isComposedBy(?x1, ?wbp1) \wedge XCoordinate(?x2) \wedge isComposedBy(?x2, ?wbp2) \wedge isGreaterThan(?x2, ?x1) \rightarrow isAfter(?hwpu2, ?hwpu1)$
The welding program unit for the horizontal welding seam whose Y-coordinate of the beginning point is smaller shall be executed before the program unit for the welding seam which possesses a larger Y-coordinate of the beginning point if the X-coordinates of both welding horizontal welding seams are equal.	$HorizontalWeldingSeam(?hws1) \wedge HorizontalWeldingSeam(?hws2) \wedge WeldBeginningPoint(?wbp1) \wedge WeldBeginningPoint(?wbp2) \wedge isComposedBy(?wbp1, ?hws1) \wedge isComposedBy(?wbp2, ?hws2) \wedge HorizontalWeldingProcess(?hwps1) \wedge HorizontalWeldingProcess(?hwps2) \wedge isDecidedBy(?hwps1, ?hws1) \wedge isDecidedBy(?hwps2, ?hws2) \wedge HorizontalWeldingProgramUnit(?hwpu1) \wedge HorizontalWeldingProgramUnit(?hwpu2) \wedge RoboticWeldingSystem(?rws) \wedge WeldingProgram(?wpm) \wedge WeldingProcess(?wps) \wedge isComposedBy(?hwpu1, ?wpm) \wedge isComposedBy(?hwpu2, ?wpm) \wedge isComposedBy(?hwps1, ?wps) \wedge isComposedBy(?hwps2, ?wps) \wedge isControlledBy(?rws, ?wpm) \wedge isRealizedBy(?wps, ?rws) \wedge isDecidedBy(?wps, ?ws) \wedge YCoordinate(?y1) \wedge isComposedBy(?y1, ?wbp1) \wedge XCoordinate(?x1) \wedge isComposedBy(?x1, ?wbp1) \wedge YCoordinate(?y2) \wedge isComposedBy(?y2, ?wbp2) \wedge XCoordinate(?x2) \wedge isComposedBy(?x2, ?wbp2) \wedge isEqualTo(?x1, ?x2) \wedge isGreaterThan(?y2, ?y1) \rightarrow isAfter(?hwpu2, ?hwpu1)$

1 4. Case study

2 Currently, the requirements of the shipbuilding industry for welding processes with higher quality
3 and greater speed are increasing dramatically. One of the most challenging tasks for implementing
4 robotic manufacturing systems in shipbuilding is that the variation of products often leads to
5 different robotic programs, even though the products are quite similar. In this study, the authors'
6 research team was required to develop a robotic welding system by a local shipbuilding company
7 for performing automated welding tasks. To further understand customer needs, the authors'
8 research team conducted a number of interviews with managers, engineers, and workers from the
9 shipbuilding company. Table 5 presents some of the specific challenges encountered in achieving
10 the automated welding process, and the solutions provided to address the challenges.

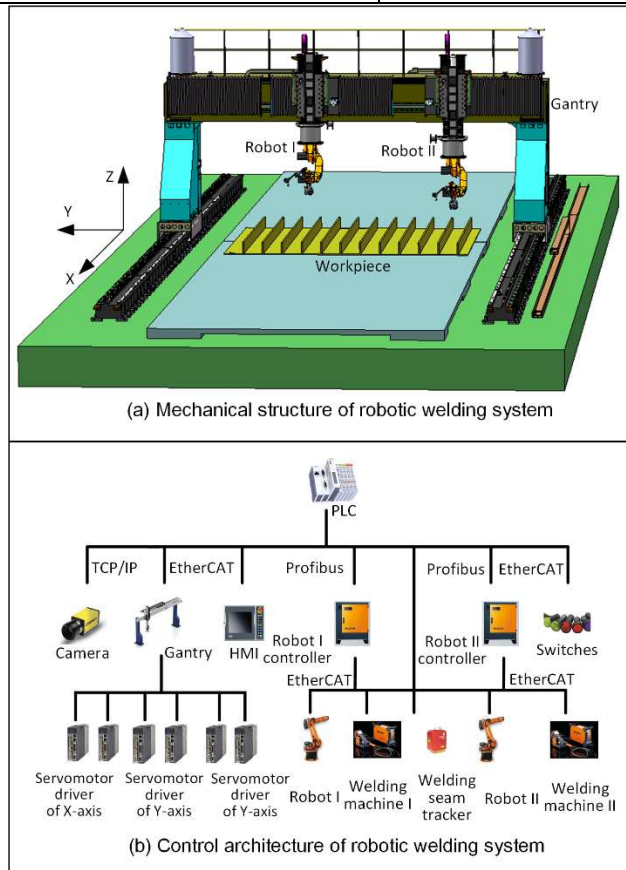
11 Considering the solutions provided for satisfying the customer requirements in achieving robotic
12 welding for shipbuilding, the authors proposed a robotic welding system whose mechanical
13 structure included two robots suspended by a gantry. Fig. 8(a) presents the mechanical structure of
14 the robotic welding system. Robots with a welding torch could be moved by the gantry, greatly
15 extending the reachability of the welding torch. In addition, the two robots could not only conduct
16 two simple independent welding tasks simultaneously, but could also collaborate with each other
17 for larger and more complex welding tasks, greatly increasing the welding efficiency.

18

1

2 Table 5 Challenges and solutions for automated welding tasks

Challenges in automated welding process for shipbuilding	Solutions for automated welding tasks
Various welding types, e.g., vertical welding, horizontal welding, fillet welding, and spot welding, exist in shipbuilding process, but the batch sizes are sometimes quite small.	The off-line programming approach is adopted to generate robot program for different welding types based on the CAD models of workpieces.
There are a large number of workpieces which should be welded, so the welding efficiency plays an important role for shipbuilding.	A multi-robots-based welding system is developed to increase the welding efficiency.
The size of the workpieces varies greatly, and the size of the largest workpiece can reach 20 m × 6 m × 4 m.	A gantry is used to move the robots and welding torches, which greatly extends the reachability of the welding torch
The errors between the final welded workpieces and the 3D models caused by the inaccuracy during the steps of workpiece preparation, blanking and pre-assembly or the thermal deformation during the welding process.	The welding beginning point, welding seam and welding ending point on the workpieces can continuously be tracked by using welding seam tracker and vision-based techniques during automatic welding process.
The safety during the welding process should be ensured.	Various collision detection sensors and algorithms have been developed to ensure the safety of welding process.
A seamless coordination among the industrial robot, gantry, welding machine, welding seam tracker, vision module, and collision avoidance sensor. to achieve a high productivity for the welding tasks.	The control architecture in which PLC is employed to integrate and coordinate different components.



3

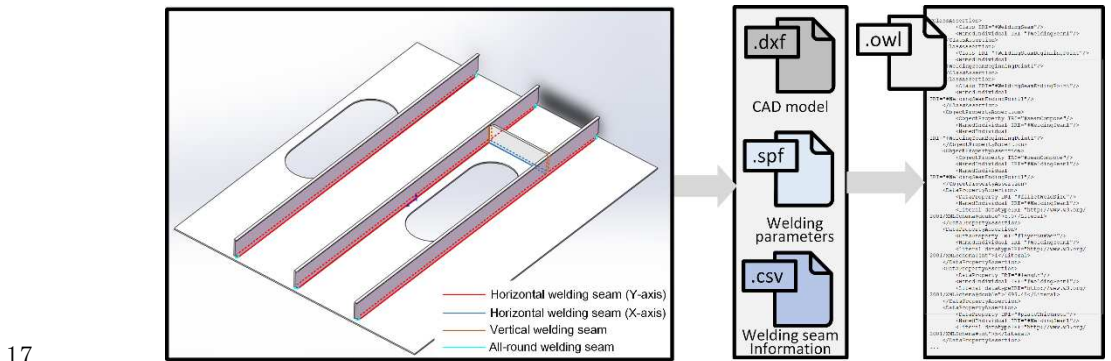
4

Fig. 8 Mechanical and control parts of robotic welding system

5

Fig. 8(b) illustrates the control architecture of the robotic welding system. A PLC was employed to

1 control the gantry, welding seam tracker, camera, switches, etc., and to coordinate them with the
 2 two robots. As a result, programming for the robotic welding system was divided into two parts,
 3 i.e., PLC programming and robot programming.
 4 Based on the proposed automatic programming generation approach, the ontological knowledge
 5 model was instantiated by using the data relating to the workpieces, robotic welding system,
 6 welding process, and basic instruction units of the programs for the robots and PLCs. Fig. 9
 7 presents the instantiation step for the workpiece product data of the ontological knowledge model.
 8 AVEVA Marine⁵ software for shipbuilding and marine design was adopted to model the
 9 workpieces, through which the workpiece product data was exported as three files, i.e., drawing
 10 exchange format (DXF), ShadowProtect (SPF), and comma-separated value (.csv) files. The DXF
 11 file contained the CAD data of the workpieces, the SPF file stored the parameters of the welding
 12 seams (such as the ID, type, length, beginning and ending points of the welding seams, and
 13 number of required welding layers), and the CSV file stored the information of the two pieces
 14 joined together to form the welding seam. To provide an explicit semantic description of the
 15 product data of workpieces, the data stored in the three files was extracted and transformed into
 16 the OWL based on the ontological knowledge model.

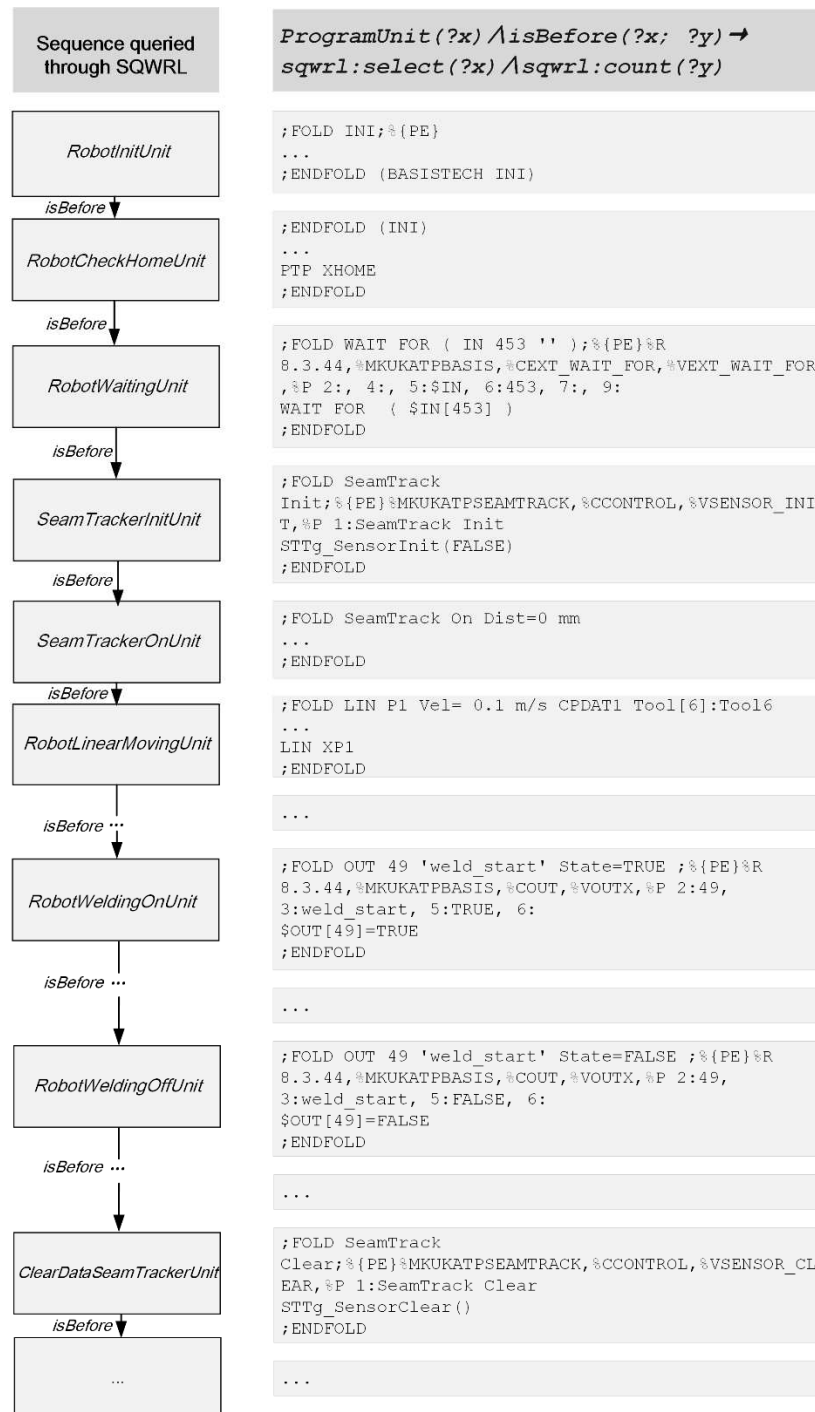


17
 18 Fig. 9 Instantiation of part of workpiece product data in ontological knowledge model

19 Similarly, the remaining parts of the proposed ontological knowledge model were instantiated by
 20 using the data relating to the robotic welding systems, welding processes, and basic instruction
 21 units of the programs for the robots and PLC. The final program was generated by combining the
 22 basic instruction units in a suitable sequence as inferred from the inter- and intra-domain
 23 relationships. For example, the welding seam tracker should be initialized after it receives the
 24 signal indicating that the gantry has arrived at the starting point of the welding seam, and before it
 25 is switched on for the following tracking process. Therefore, the intra-domain relationships among
 26 the *SeamTrackerInitUnit*, *RobotWaitingUnit*, and *SeamTrackerOnUnit* can be summarized as
 27 follows: *SeamTrackerInitUnit* should be conducted after *RobotWaitingUnit*, and before
 28 *SeamTrackerOnUnit*. In addition, the inter-domain relationships were also considered for program
 29 generation. For example, a horizontal welding seam whose length exceeds the reach of an
 30 industrial robot should be welded by using the gantry, as controlled by the PLC. As a result, the
 31 *RobotWeldingOffUnit* should be executed after receiving the PLC signal indicating that the
 32 welding torch has arrived at the end point of the welding seam. Based on the semantic description
 33 of the existing data and reasoning mechanism, the inference engine Pellet was used to infer the

⁵ <https://sw.aveva.com/marine>

1 sequence of the basic instruction units of the programs for robots or PLCs, and the results could be
 2 queried through the SQWRL. Fig. 10 presents the robot program for welding the first horizontal
 3 welding seam whose length exceeds the reach of the industrial robot. The generated robot program
 4 was first validated by simulation in DELMIA Robotics (Fig. 11(a)), and then was imported into
 5 the robotic welding system (Fig. 11(b and c)). Fig. 11(d-f) show the horizontal, all-round, and
 6 vertical welding processes achieved by the generated robot program. The quality testing results of
 7 the three welds indicate that the welding quality and stability are ensured by standardizing the
 8 rules and knowledge related to previous successful welding processes. The PLC program can be
 9 obtained by performing similar generation steps.



1

Fig. 10 Automatic program generation for industrial robot



2

3

Fig. 11 Experimental validation of generated robot program

4 5. Discussion

5 To make changes to a product or service to satisfy a given customer group, current manufacturing
6 companies generally pay more attention to mass customization, i.e., to develop various products
7 for different customers. The shipbuilding industry presented in Section 4 is considered as a typical
8 example of mass customization production, where various welding types are involved in
9 shipbuilding processes but the batch sizes are sometimes quite small; this may result in diversity in
10 program adjustments, even though many of the sections of a specific program may not be changed.
11 Accordingly, a KBE automatic program-generation approach was developed to solve the
12 aforementioned problem. The proposed approach cannot only achieve automatic generation for
13 manufacturing programs for different workpieces without human interaction, but also supports the
14 reuse of existing manufacturing cases by standardizing the manufacturing components, processes,
15 and programs. The main contributions of this study are as follows.

- 1 • Explicit semantic descriptions are defined for concepts and their relationships in the domains
2 of product models of workpieces, robotic manufacturing processes, robotic manufacturing
3 systems, and robotic manufacturing programs; these can be easily understood by users and
4 processed by computers.
- 5 • The intra- and inter-domain relationships are described by the SWRL. Based on these, the
6 reasoning mechanism of the knowledge model is established to infer the sequences of the
7 basic instruction units in the required robotic manufacturing program.
- 8 • Different program engineers may develop different programs for the same manufacturing
9 case, which cannot ensure programming efficiency and production quality. The proposed
10 automatic program generation approach provides effective support for the standardization of
11 the rules and knowledge related to manufacturing programs that have been proven by
12 previous successful manufacturing cases, and can thereby greatly improve the manufacturing
13 stability and production quality.
- 14 • Based on the semantic descriptions and reasoning mechanism of the proposed knowledge
15 model, the basic instruction units of the manufacturing program can be instantiated based on
16 data extracted from product models and integrated according to their relationships as inferred
17 by the reasoning mechanism, thereby generating the robotic manufacturing program.

18 Although the effectiveness of the automatic program generation approach was demonstrated based
19 on an industrial case for a robotic welding system, the proposed approach still needs to be further
20 expanded, from the following perspectives.

- 21 • The manufacturing program was determined by comprehensively considering the interactions
22 among the robotic manufacturing system, robotic manufacturing processes, and product data
23 of the workpieces. Therefore, the authors proposed an automatic program generation
24 approach based on four root concepts (i.e., robotic manufacturing system, process, program,
25 and product data of workpieces). The semantic relationships among the four root concepts
26 and how to utilize their interactions (i.e., inter-domain relationships) to generate the
27 manufacturing program comprise the main contributions of this study. Different domain
28 ontologies can be used to describe the details of each root concept. For example, Barbau et al.
29 proposed OntoSTEP as a geometry-enhanced ontological model for describing the concepts
30 relating to the geometric information and shape representations in the product models of
31 workpieces [60]. Stenmark and Malec presented a domain ontology for providing semantic
32 descriptions of the skills and components of robotic manufacturing systems [63]. Qiao et al.
33 focused on an ontology for defining the main concepts and relations in the domain of
34 manufacturing process planning [64]. As a result, when applying the proposed automatic
35 program generation approach, the ontological knowledge model for providing the semantic
36 descriptions for the relevant concepts can be either defined by the users, or can be obtained
37 by integrating the aforementioned existing domain ontologies using ontology matching
38 techniques. Admittedly, with developments in technology, neither method can contain all
39 concepts related to robotic manufacturing. Therefore, the concepts and relationships of
40 certain specific manufacturing domains can be defined and integrated into the ontological
41 knowledge model.

- 1 • The proposed knowledge-based program generation approach was developed based on an
2 offline programming paradigm. According to the offline programming paradigm, the program
3 for the robotic manufacturing systems is automatically generated by considering the product
4 data of the workpieces, i.e., without adopting a real robot. Therefore, uncertainties in the
5 environment, such as changes and adjustments occurring during the manufacturing process,
6 are not considered when generating the manufacturing program. Such environmental
7 uncertainty is considered difficult to integrate into the automatic program generation process
8 for the following four reasons. First, environmental uncertainty is irregular, and often occurs
9 stochastically. Second, environmental uncertainty is non-uniform, as each type of
10 environmental uncertainty can be different. Third, environmental uncertainty is
11 non-deterministic, as any environmental uncertainty may change the state of the system,
12 potentially leading to another uncertainty. Finally, environmental uncertainty is not
13 standardized, and may require different solutions. As a result, it is challenging to provide an
14 explicit semantic description of various environmental uncertainties or to construct a
15 rule-based reasoning mechanism to determine a suitable response. Current robotic
16 manufacturing systems always adopt sensors or specific techniques to detect and respond to
17 environmental uncertainties. Taking a robotic welding system as an example, an axis-aligned
18 bounding box-based (AABB) algorithm can be used to detect collisions between the robot
19 and the environment during the welding process [65], and a welding seam tracker can be
20 adopted to detect the deformation of the workpiece and/or to instruct the robot to follow the
21 actual welding seam [66]. These instructions are not added to the manufacturing program, but
22 are sent to the robot controller, PLCs, and other real-time controllers directly, so as to ensure
23 the real-time response to environmental uncertainties. Recently, some data exchange
24 standards, such as MTConnect⁶ and the OPC Unified Architecture (OPC UA)⁷, have been
25 proposed to achieve data communication, integration, and management in manufacturing
26 systems [67]. Researchers have already attempted to combine semantic web technologies
27 with MTConnect [68] and OPC UA [69] by adding semantic values to the data
28 representations in the above-mentioned two standards. Considering the diverse data relating
29 to the environmental uncertainties that can be captured by sensors or other data acquisition
30 systems, these semantic standards can be used as potential solutions for converting the
31 non-uniform environment data into a machine-interpretable ontology. This can potentially
32 enable interlinking between environmental uncertainties and concepts relating to robotic
33 manufacturing systems and product data of workpieces, such that a manufacturing program
34 considering environmental uncertainties can be generated.
- 35 • A rule-based reasoning mechanism was proposed to generate the manufacturing program in
36 this study, and an industrial case based on a robotic welding system was adopted to
37 demonstrate the effectiveness of the proposed approach. However, there are potentially
38 several alternative combinations of basic instruction units, and all of them may satisfy the
39 predefined rules if the number of basic instruction units is large or if the rules restricting the
40 sequence of instruction units are loose. Therefore, users should select the most suitable
41 combinations among the alternatives by considering other requirements or predefined key

⁶ <http://www.mtconnect.org/>

⁷ <https://opcfoundation.org/about/opc-technologies/opc-ua>

1 performance indicators, such as manufacturing efficiency, power consumption, and operator
2 fatigue. A ranking method is considered an effective support to help users evaluate
3 alternatives. Current ranking methods, such as the multi-attribute utility analysis [70], fuzzy
4 methods [71], and analytic hierarchy process [72], can provide a comprehensive index value
5 for each alternative for ranking. The authors have also proposed several multi-attribute
6 decision-making models [73,74] to help users rank alternatives. However, a considerable
7 amount of effort is still needed to develop a systematic methodology for integrating the
8 rule-based reasoning mechanism with a ranking method in an expert system, so as to generate
9 and evaluate alternative programs for optimal decisions.

10 **6. Conclusion**

11 Considering the limitations of current programming approaches for robotic manufacturing systems,
12 a knowledge-based approach to help users achieve automatic program generation is presented in
13 this paper. An ontological knowledge model is developed to provide an explicit semantic
14 description of the relevant concepts of the robotic manufacturing system, basic instruction units of
15 the program, and product models of the workpieces. A rule-based reasoning mechanism is
16 established to infer the implicit relationships between the basic instruction units of the
17 manufacturing program. Based on the semantic descriptions, the basic instruction units can be
18 integrated to generate the required robotic manufacturing program according to the relationships
19 inferred by the reasoning mechanism. A case study based on a robotic welding system is
20 adopted to demonstrate the effectiveness of the proposed automatic program generation
21 approach.

22 **7. Acknowledgements**

23 This project is supported by the National Natural Science Foundation of China (Grant No.
24 51805437), the Natural Science Foundation of Shaanxi Province (Grant No. 2020JQ-187), the
25 Fundamental Research Funds for the Central Universities (Grant No. G2018KY0301) and the
26 National Defense Basic Scientific Research Program of China (Grant No. JCKY2018607C004).

27 **References**

- 28 [1] H. Chen, W. Sheng, Transformative CAD based industrial robot program generation, *Robot.*
29 *Comput. Integr. Manuf.* 27 (2011) 942–948.
- 30 [2] R.Y. Zhong, Q.Y. Dai, T. Qu, G.J. Hu, G.Q. Huang, RFID-enabled real-time manufacturing
31 execution system for mass-customization production, *Robot. Comput. Integr. Manuf.* 29 (2013)
32 283–292.
- 33 [3] C. Zheng, X. Qin, B. Eynard, J. Bai, J. Li, Y. Zhang, SME-oriented flexible design approach
34 for robotic manufacturing systems, *J. Manuf. Syst.* 53 (2019) 62–74.
- 35 [4] D.J. Medeiros, R.P. Sadowski, Simulation of robotic manufacturing cells: a modular approach,
36 *Simulation.* 40 (1983) 3–12.
- 37 [5] J. Wang, C. Xu, J. Zhang, J. Bao, R. Zhong, A collaborative architecture of the industrial

- 1 internet platform for manufacturing systems, *Robot. Comput. Integr. Manuf.* 61 (2020) 101854.
- 2 [6] M. Song, T.J. Tarn, N. Xi, Integration of task scheduling, action planning, and control in
3 robotic manufacturing systems, *Proc. IEEE.* 88 (2000) 1097–1107.
- 4 [7] Z.M. Bi, Y. Lin, W.. J. Zhang, The general architecture of adaptive robotic systems for
5 manufacturing applications, *Robot. Comput. Integr. Manuf.* 26 (2010) 461–470.
- 6 [8] Y. Alsafi, V. Vyatkin, Ontology-based reconfiguration agent for intelligent mechatronic
7 systems in flexible manufacturing, *Robot. Comput. Integr. Manuf.* 26 (2010) 381–391.
- 8 [9] P. Wegner, Concepts and paradigms of object-oriented programming, *ACM Sigplan Oops
9 Messenger.* 1 (1990) 7–78.
- 10 [10] L. Lin, M. Wakabayashi, S. Adiga, Object-oriented modeling and implementation of control
11 software for a robotic flexible manufacturing cell, *Robot. Comput. Integr. Manuf.* 11 (1994) 1–
12 12.
- 13 [11] P. O’Grady, R. Seshadri, X-cell — intelligent cell control using object-oriented programming
14 (Part I), *Robot. Comput. Integr. Manuf.* 4 (1991) 157–163.
- 15 [12] A. Fahim, K. Choi, The UniSet approach for the programming of flexible manufacturing cells,
16 *Robot. Comput. Integr. Manuf.* 14 (1998) 69–78.
- 17 [13] E. Argente, V. Julian, V. Botti, Multi-agent system development based on organizations,
18 *Electron. Notes Theor. Comput. Sci.* 150 (2006) 55–71.
- 19 [14] J. Pang, X. Wang, J. Peng, NC manufacturing system based on multi-agent programming, *Key
20 Eng. Mater.* 693 (2016) 1676–1683.
- 21 [15] S. Bussmann, Agent-oriented programming of manufacturing control tasks, in: *Proc. Int. Conf.
22 Multi Agent Syst., Paris, France, 1998.*
- 23 [16] S. Scholz, A. Elkaseer, M. Salem, V. Hagenmeyer, Software toolkit for visualization
24 and process selection for modular scalable manufacturing of 3D micro-devices, in: *40th Anniv.
25 Int. Conf. Inf. Syst. Archit. Technol., Wrocław, Poland, 2019.*
- 26 [17] T. AlGeddawy, A new model of modular automation programming in changeable
27 manufacturing systems, in: *27th Int. Conf. Flex. Autom. Intell. Manuf., Modena, Italy, 2017.*
- 28 [18] International Electrotechnical Commission, IEC TC65/WG6, Function blocks for industrial
29 process measurement and control systems-Part 1: Architecture, 1999.
- 30 [19] X.W. Xu, L. Wang, Y. Rong, STEP-NC and function blocks for interoperable manufacturing,
31 *IEEE Trans. Autom. Sci. Eng.* 3 (2006) 297–308.
- 32 [20] L. Wang, G. Adamson, M. Holm, P. Moore, A review of function blocks for process planning
33 and control of manufacturing equipment, *J. Manuf. Syst.* 31 (2012) 269–279.
- 34 [21] B. Yao, Z. Zhou, L. Wang, W. Xu, J. Yan, Q. Liu, A function block based cyber-physical
35 production system for physical human–robot interaction, *J. Manuf. Syst.* 48 (2018) 12–23.
- 36 [22] J.N. Pires, J.M.G.S. da Costa, Object-oriented and distributed approach for programming
37 robotic manufacturing cells, *Robot. Comput. Integr. Manuf.* 16 (2000) 29–42.
- 38 [23] M.R. Pedersen, L. Nalpantidis, R.S. Andersen, C. Schou, S. Bøgh, V. Krüger, O. Madsen,
39 Robot skills for manufacturing: from concept to industrial deployment, *Robot. Comput. Integr.
40 Manuf.* 37 (2016) 282–291.
- 41 [24] M.P. Deisenroth, K.K. Krishnan, On-line programming, in: *Handb. Ind. Robot., John Wiley &
42 Sons, Hoboken, USA, 1999.*
- 43 [25] C. Kohrt, R. Stamp, A.G. Pipe, J. Kiely, G. Schiedermeier, An online robot trajectory planning
44 and programming support system for industrial use, *Robot. Comput. Integr. Manuf.* 29 (2013)

- 1 71–79.
- 2 [26] H. Zhang, H. Chen, N. Xi, G. Zhang, J. He, On-line path generation for robotic deburring of
3 cast aluminum wheels, in: *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Beijing, China, 2006.
- 4 [27] B. Hein, M. Hensel, H. Worn, Intuitive and model-based on-line programming of industrial
5 robots: a modular on-line programming environment, in: *IEEE Int. Conf. Robot. Autom.*,
6 Pasadena, USA, 2008.
- 7 [28] P. Neto, J.N. Pires, A.P. Moreira, Accelerometer-based control of an industrial robotic arm, in:
8 *18th IEEE Int. Symp. Robot Hum. Interact. Commun.*, Toyama, Japan, 2009.
- 9 [29] V.S. Bottazzi, J.F.C. Fonseca, Off-Line robot programming framework, in: *Jt. Int. Conf. Auton.*
10 *Auton. Syst. Int. Conf. Netw. Serv.*, Papeete, Tahiti, French Polynesia, 2005.
- 11 [30] Z. Liu, W. Bu, J. Tan, Motion navigation for arc welding robots based on feature mapping in a
12 simulation environment, *Robot. Comput. Integr. Manuf.* 26 (2010) 137–144.
- 13 [31] C.S. Kim, K.S. Hong, H.Y.S. Han, S.H. Kim, S.C. Know, PC-based off-line programming
14 using VRML for welding robots in shipbuilding, in: *Proc. IEEE Conf. Robot. Autom.*
15 *Mechatronics*, Singapore, Singapore, 2004.
- 16 [32] S. Zivanovic, N. Slavkovic, D. Milutinovic, An approach for applying STEP-NC in robot
17 machining, *Robot. Comput. Integr. Manuf.* 49 (2018) 361–373.
- 18 [33] D. Ding, C. Shen, Z. Pan, D. Cuiuri, H. Li, N. Larkin, S. van Duin, Towards an automated
19 robotic arc-welding-based additive manufacturing system from CAD to finished part, *Comput.*
20 *Des. Des.* 73 (2016) 66–75.
- 21 [34] T. Lu, G.C.I. Lin, An on-line relative position and orientation error calibration methodology for
22 workcell robot operations, *Robot. Comput. Integr. Manuf.* 13 (1997) 89–99.
- 23 [35] J. Cortsen, J.A. Jørgensen, D. Sølvason, H.G. Petersen, Simulating robot handling of large
24 scale deformable objects: manufacturing of unique concrete reinforcement structures, in: *IEEE*
25 *Int. Conf. Robot. Autom.*, Saint Paul, USA, 2002.
- 26 [36] S.F. Chan, R. Kwan, Post-processing methodologies for off-line robot programming within
27 computer integrated manufacture, *J. Mater. Process. Technol.* 139 (2003) 8–14.
- 28 [37] Z. Pei, L. Chen, Welding component identification and solder joint inspection of automobile
29 door panel based on machine vision, in: *Chinese Control Decis. Conf.*, Shenyang, China, 2018.
- 30 [38] Z. Jia, T. Wang, L. Li, J. He, Laser vision-based automatic trajectory planning technology for
31 spatial intersecting joint weld, *Int. J. Precis. Eng. Manuf.* 21 (2020) 45–55.
- 32 [39] L. Yang, E. Li, T. Long, J. Fan, Z. Liang, A novel 3-D path extraction method for arc welding
33 robot based on stereo structured light sensor, *IEEE Sens. J.* 19 (2019) 763–773.
- 34 [40] Z.M. Bi, Z. Miao, B. Zhang, C.W. Zhang, The state of the art of testing standards for integrated
35 robotic systems, *Robot. Comput. Integr. Manuf.* 63 (2020) 101893.
- 36 [41] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A.Y. Ng, ROS: an
37 open-source Robot Operating System, *ICRA Work. Open Source Softw.* 3 (2009) 5–10.
- 38 [42] J. Kerr, K. Nickels, Robot operating systems: Bridging the gap between human and robot, in:
39 *44th Southeast. Symp. Syst. Theory*, Jacksonville, FL, USA, 2012: pp. 99–104.
- 40 [43] V. Mayoral-Vilches, M. Pinzger, S. Rass, B. Dieber, E. Gil-Uriarte, Can ROS be used securely
41 in industry? *Red teaming ROS-Industrial*, 2020.
- 42 [44] T. Segaran, C. Evans, J. Taylor, *Programming the semantic web*, O'Reilly Media,
43 Sebastopol, USA, 2009.
- 44 [45] R. Costa, C. Lima, J. Sarraipa, R. Jardim-Gonçalves, Facilitating knowledge sharing and reuse

- 1 in building and construction domain: an ontology-based approach, *J. Intell. Manuf.* 27 (2016)
2 263–282.
- 3 [46] H.K. Tönshoff, E. Menzel, H.S. Park, A knowledge-based system for automated assembly
4 planning, *CIRP Ann.* 41 (1992) 19–24.
- 5 [47] T. Sadek, R. Theiss, Knowledge based assistance for conceptual development of industrial
6 Product-Service Systems, *Adv. Intell. Soft Comput.* 66 (2010) 1647–1664.
- 7 [48] S.M. Sapuan, H.S. Abdalla, A prototype knowledge-based system for the material selection of
8 polymeric-based composites for automotive components, *Compos. Part A Appl. Sci. Manuf.* 29
9 (1998) 731–742.
- 10 [49] A. Olivares-Alarcos, D. Beßler, A. Khamis, P. Goncalves, M.K. Habib, J. Bermejo-Alonso, M.
11 Barreto, M. Diab, J. Rosell, J. Quintas, J. Olszewska, H. Nakawala, E. Pignaton, A. Gyrard, S.
12 Borgo, G. Alenyà, M. Beetz, H. Li, A review and comparison of ontology-based approaches to
13 robot autonomy, *Knowl. Eng. Rev.* 34 (2020) 1–38.
- 14 [50] M. Stenmark, J. Malec, K. Nilsson, A. Robertsson, On distributed knowledge bases for
15 robotized small-batch assembly, *IEEE Trans. Autom. Sci. Eng.* 12 (2015) 519–528.
- 16 [51] M. Stenmark, J. Malec, Knowledge-based instruction of manipulation tasks for industrial
17 robotics, *Robot. Comput. Integr. Manuf.* 33 (2015) 56–67.
- 18 [52] R. Patel, M. Hedelind, P. Lozan-Villegas, Enabling robots in small-part assembly lines: the
19 “ROSETTA approach” - an industrial perspective, in: *7th Ger. Conf. Robot.*, Munich, Germany,
20 2012: pp. 279–283.
- 21 [53] M. Tenorth, G. Bartels, M. Beetz, Knowledge-based specification of robot motions, in: *21st*
22 *Eur. Conf. Artif. Intell.*, Prague, Czech Republic, 2014: pp. 873–878.
- 23 [54] S.R. Fiorini, J.L. Carbonera, P. Gonçalves, V.A. Jorge, V.F. Rey, T. Haidegger, M. Abel, S.A.
24 Redfield, S. Balakirsky, V. Ragavan, H. Li, Extensions to the core ontology for robotics and
25 automation, *Robot. Comput. Integr. Manuf.* 33 (2015) 3–11.
- 26 [55] V.A. Jorge, V.F. Rey, R. Maffei, S.R. Fiorini, J.L. Carbonera, F. Branchi, J.P. Meireles, G.S.
27 Franco, F. Farina, T.S. Da Silva, M. Kolberg, Exploring the IEEE ontology for robotics and
28 automation for heterogeneous agent interaction, *Robot. Comput. Integr. Manuf.* 33 (2015) 12–
29 20.
- 30 [56] W. Shan, L. Dongbo, J. Gao, L. Jing, A knowledge based machine tool maintenance planning
31 system using case-based reasoning techniques, *Robot. Comput. Integr. Manuf.* 58 (2019) 80–
32 96.
- 33 [57] SCRA STEP Application, Step application handbook ISO 10303 Version 3, Hand Book, 2006.
- 34 [58] S.J. Fennes, S. Foufou, C. Bock, R.D. Sriram, CPM : a core model for product data, *J. Comput.*
35 *Inf. Sci. Eng.* (2006) 1–14.
- 36 [59] F. Noël, L. Roucoules, The PPO design model with respect to digital enterprise technologies
37 among product life cycle, *Int. J. Comput. Integr. Manuf.* 21 (2008) 139–145.
- 38 [60] R. Barbau, S. Kríma, S. Rachuri, A. Narayanan, X. Fiorentini, S. Foufou, R.D. Sriram,
39 *OntoSTEP: Enriching product model data using ontologies*, *Comput. Des.* 44 (2012) 575–590.
- 40 [61] H. Panetto, M. Dassisti, A. Tursi, ONTO-PDM: Product-driven ONTOlogy for Product Data
41 Management interoperability within manufacturing process environment, *Adv. Eng.*
42 *Informatics.* 26 (2012) 334–348.
- 43 [62] R. De Nicola, F. Vaandrager, Action versus state based logics for transition systems, in: *LITP*
44 *Spring Sch. Theor. Comput. Sci.*, Springer, Berlin, Germany, 1990: pp. 407–419.

- 1 [63] M. Stenmark, J. Malec, Knowledge-based industrial robotics, *Front. Artif. Intell. Appl.* 257
2 (2013) 265–274.
- 3 [64] L. Qiao, Y. Qie, Z. Zhu, Y. Zhu, U.K. Zaman, N. Anwer, An ontology-based modelling and
4 reasoning framework for assembly sequence planning, *Int. J. Adv. Manuf. Technol.* 94 (2018)
5 4187–4197.
- 6 [65] K. Okada, M. Inaba, H. Inoue, Real-time and precise self collision detection system for
7 humanoid robots, in: *IEEE Int. Conf. Robot. Autom.*, 2005: pp. 1060–1065.
- 8 [66] A. Rout, B.B.V.L. Deepak, B.B. Biswal, Advances in weld seam tracking techniques for
9 robotic welding: a review, *Robot. Comput. Integr. Manuf.* 56 (2019) 12–37.
- 10 [67] C. Liu, H. Vengayil, Y. Lu, X. Xu, A cyber-physical machine tools platform using OPC UA
11 and MTConnect, *J. Manuf. Syst.* 51 (2019) 61–74.
- 12 [68] G. Alvanou, I. Lytra, N. Petersen, An MTConnect ontology for semantic industrial machine
13 sensor analytics, *Proc. Work. Semant. Web Things Ind.* 4 (2018) 57–80.
- 14 [69] B. Katti, C. Plociennik, M. Schweitzer, SemOPC-UA: Introducing semantics to OPC-UA
15 application specific methods, *IFAC-PapersOnLine.* 51 (2018) 1230–1236.
- 16 [70] R. Malak, J. Aughenbaugh, C. Paredis, Multi-attribute utility analysis in set-based conceptual
17 design, *Comput. Des.* 41 (2009) 214–227.
- 18 [71] B. Zhu, Z. Wang, H. Yang, R. Mo, Y. Zhao, Applying fuzzy multiple attributes decision
19 making for product configuration, *J. Intell. Manuf.* 19 (2008) 591–598.
- 20 [72] R. Handfield, S. V. Walton, R. Sroufe, S.A. Melnyk, Applying environmental criteria to
21 supplier assessment: a study in the application of the Analytical Hierarchy Process, *Eur. J. Oper.*
22 *Res.* 141 (2002) 70-87.
- 23 [73] C. Zheng, X. Qin, B. Eynard, J. Li, J. Bai, Y. Zhang, S. Gomes, Interface model-based
24 configuration design of mechatronic systems for industrial manufacturing applications, *Robot.*
25 *Comput. Integr. Manuf.* 59 (2019) 373–384.
- 26 [74] Y. Zhang, Y. Xu, A. Bernard, A new decision support method for the selection of RP process:
27 knowledge value measuring, *Int. J. Comput. Integr. Manuf.* 27 (2014) 747–758.
- 28
- 29