Dynamic and unified modelling of sustainable manufacturing capability for industrial robots in cloud manufacturing

Zhao, Yuanyuan; Liu, Quan; Xu, Wenjun; Wu, Xingxing; Jiang, Xuemei; Zhou, Zude; Pham, Duc Truong

DOI:
10.1007/s00170-017-0634-1

Dynamic and Unified Modeling of Sustainable Manufacturing Capability for Industrial Robots in Cloud Manufacturing

Yuanyuan Zhao[a,b], Quan Liu[a,c], Wenjun Xu[a,b]*, Xingxing Wu[a,b], Xuemei Jiang[a,c], Zude Zhou[a,c], Duc Truong Pham[d]

[a] School of Information Engineering, Wuhan University of Technology, Luoshi Road 122, Wuhan 430070, China

[b] Hubei Key Laboratory of Broadband Wireless Communication and Sensor Networks (Wuhan University of Technology), Wuhan 430070, China

[c] Key Laboratory of Fiber Optic Sensing Technology and Information Processing (Wuhan University of Technology), Ministry of Education, Luoshi Road 122, Wuhan 430070, China

[d] Department of Mechanical Engineering, School of Engineering, University of Birmingham, Birmingham B15 2TT, UK

*Corresponding author

Email address: xuwenjun@whut.edu.cn

Abstract: Industrial robots (IRs) are the important driving force to enable the production activities more automotive and high-efficient in modern manufacturing systems. However, in order to realize the effective employment and intelligent configuration of IRs in cloud manufacturing environment, it is required that the sustainable manufacturing capabilities of IRs can be described in a unified and formal manner. In this paper, a unified sustainable manufacturing capability (SMC) of IRs model is constructed in terms of functional attributes, structural information, activities and process condition. A hybrid logic description method integrating OWL with dynamic description logic (DLL) is adopted to provide semantical representation to both the static and dynamic characteristics of SMC. An interval-state description method is proposed to present energy consumption during IR's process in sections. Based on the constructed model, three types' rules are defined to reason the capability of IRs, including stability, energy consumption and production capacity. Finally, a cloud-based prototype system architecture is illustrated. An IR service platform is developed and implemented to verify the proposed model and the defined rules.

Keywords: industrial robots, sustainable manufacturing capability, dynamic and unified modeling, ontology, dynamic description logic

1. Introduction

Cloud manufacturing promotes the collaboration and interoperability of industry production due to its characteristics, including cloud-based, service-oriented, intelligence [1, 2]. It provides a sharable pool for distributed resources and releases them as manufacturing services. In this pool, both hardware and software resources/services can be combined and configured automatically in digital form under virtual environment through internet[3, 4, 5]. This mechanism facilitates in large extent collaboration among multiple enterprises ignoring territory limitation, and allows effective business interactions over internet. Meanwhile, service oriented architecture supplies interoperable form to resources capability so that the related data can be exchanged among different manufacturing activities, for example product design, process planning and machining. Hence, cloud-based mode is significant to manufacturing industry. However, in such kind of environment, resources and service are disseminated and exchanged by machine over the Internet. Only ensuring the correct comprehension and communication of them among machines can indeed activate the mode. Therefore, it is necessary and fundamental to provide veracious and machine-readable description to resources and services

Moreover, from the development requirement of industry, sustainability is a main demand that manufacturers have to pay attention to in the present. Domain experts keep seeking suitable solutions to the environmental and social issues that are caused by the development of industrial production. In practical manufacturing applications, energy consumption during machining process is one of key factors that influence environment significantly. From both economic and ecological perspectives, energy consumption should be considered as one of the optimization objectives for configuration and selection of resources and services in various manufacturing activities [6, 7, 8]. Efficient and energy-saving usage of resources is imperative for the sustainable progress of manufacturing. And it requires appraising resource capability from multiple respects, not only relating to nominal technical specification, but also involving the real process status. Hence, classical function statement is not sufficient to current industrial applications.

For modern manufacturing industry, another cordial is the employment of Industrial robots (IRs). As a type of automated equipment, it updates manufacturing automation level in a great extent. IRs are able to supply many skills, e.g. strength, precision and sensing, which surpassing those of humans. Robotic manipulator is regarded as an immediate viable alternative tool for repetitive operations, such as routine assembly line, to free human from boring and heavy works [9,10,11]. As a result, IR's application involves a broad range of industrial field, including material transfer, precision assembly, welding, and machining. It is undoubtable that robots would be main steam in future manufacturing. However, IRs, as a kind of electrical device, expend vast electricity during industrial production, which cannot be neglected for the saving of environment and energy [12,13]. From the respect of sustainability, utilizing robots in an energy-efficient way conforms to the development of industry, and it is also a key issue that engineers and experts eager to solve. Hence, for the manufacturing applications in cloud-based environment, IRs capability description should satisfy the need of effective and energy-efficient utilization.

So far, many description models have already been constructed to represent robot and robotic systems, such as URDF[14], IRDM [15], SRDL[16], CORA[17], etc. These models focus on modelling robots from its components, kinematic structure and actions to express robot capacity for respective targets. URDF is the kinematics and dynamics of robots model that is built by describing structural characteristics, such as links and joints. IRDM provides STEP-compliant data model for the exchange of CAX and robot off-line programming systems. These models offer detailed description for robot, however they lack semantic information. SRDL specifies industrial robots with its capabilities and actions in semantic pattern. CORA is an upper ontology for all types of robots. They describe robots' capability only from the respects of components and dynamic structural kinematics, without considering the status of IRs and energy consumption during use. For the energy consumption modelling of IRs, although there are also many researches on it, most of them are mathematic models built by relevant parameters for computing [13, 18, 19]. But for intelligent configuration in cloud environment, it is necessary to build energy models in logic semantic pattern to enable reasoning for decision making. From all the above models, we can find out that there is not a description frame that can combine them and present them in a unified format.

Ontology is a popular semantic modelling method for intelligent applications, providing a formal and explicit representation to the facts of world. By using description logic language, ontology enables the consistency reasoning of concepts to avoid expression conflicts. OWL (Ontology Web Language) is extensional description logic language, recommended by World Wide Web Consortium (W3C). As an ontology language, it offers formal syntax and semantics to develop ontological models within a domain of interest in terms of concepts, properties, relationships, logic constraints and axioms. However description logics were originally designed for representing only static knowledge, it has description limitation on the changes in time or under certain actions [20, 21]. In our application, in order to satisfy the demand of IRs' intelligent usage in manufacturing, sustainable manufacturing capability (SMC) modeling of IRs is desired, which involves notions that are of dynamic feature, such as actions, operations, status, etc. Obviously, only OWL is insufficient..

Hence, In this paper, a unified sustainable manufacturing capability (SMC) model of IRs is constructed in terms of functional attributes, structural information, activities and process condition. A hybrid logic description method integrating OWL with dynamic description logic (DLL) is adopted to provide semantical representation to both the static and dynamic characteristics of SMC model. Through this compound language, a fundamental ontology is built by OWL and DDL is used to define the actions of IRs. For the energy consumption state during process, an interval-state description method is proposed to present energy consumption in sections. Based on the constructed model, three types' rules are defined to reason the capability of IRs, including stability, energy consumption and production capacity.

The rest of this paper is organized as follows: In the next section, the literatures related to the modelling of robots are summarized. Afterwards, a unified description framework for the SMC of IRs is proposed. In section 4, the semantic description method is explained containing fundamental ontology, activities and energy consumption representation.. Based on these description, ,three types rules are defined for the IRs' capability reasoning are expressed in section 5. Then the developed method is verified by a prototype system in section 6. Finally, the conclusions and future works are presented in section 7.

2. Related Works

With the extensive utilization of robots in the field of industry, robots and robotic systems modelling are studied to improve performance or address capacity under different application circumstances. Most of them construct structure, component, kinematic and dynamics models to describe functional parameters and behaviours. A structural model of robot was present in [9], and the parameters focusing on the analysis of the system's stiffness and its behaviour during the milling process were identified. A multi-body dynamic model of a serial robot was elaborated by using beam elements, including geometry, elasticity, and damping parameters, which can be adjusted to optimal machining trajectory planning in [22]. A structural model to control kinematic dynamic of robots is developed for scheduling robust by two modelling methods: single joint modelling and multiple joints modelling [23]. A model driven approach for standardizing robotic system that is composed of sensors, actuators, auxiliary elements, tools etc. which are incorporated into robot was proposed in [24]. A representation schema for tolerances associated with robotic machine operations that are to load and unload a machine was built in [25]. These models describe robot capability and performance only from the robot's structural characteristics, functional components, and inherent parameters, without considering the status of robot during the course of working. Meanwhile, these models are diverse due to different application targets.

For unified modelling of robots, some general models have already been constructed. Universal Robotic Description Format (URDF) [14] specifies the kinematics and dynamics, the visual representation and collision model of a robot by using primarily two different language elements, namely links and joints. URDF file follows XML format to describe a robot, which provides a unified model to a flexible framework, named ROS(Robot Operation System). This framework is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behaviour across a wide variety of robotic platform. IRDM [15] is a STEP-compliant industrial robot data model that combine existing related model and standards, including UMRM (Unified Manufacturing Resource Model) [26], ISO10303-105 and ISO 14649.The information content is categorized into product model, process model, resource model and mathematic model. In IRDM, the industrial robot resources (e.g. kinematic configurations, end effectors, auxiliary devices etc.) and the kinematic and dynamic aspects of a robot are represented using STEP EXPRESS schema. In addition, using the object-oriented concept of STEP-NC, robot manufacturing tasks are defined as projects, work plans and working steps. The developed data model not only benefits robot simulation systems, but also for the improvement of traditional industrial robot controllers. Although these two models contribute to the unified and formal description of robots, they lack semantic information that is essential to unambiguous searching and sharing over the internet. Meanwhile, neither XML format nor EXPRESS enables intelligent reasoning for checking representation consistency of terminologies due to lacking logic description.

In the field of robot, ontology is also applied to robot modeling. CORA is a core ontology for robotic and automation developed by an official IEEE Working Group, named Ontologies for Robotics and Automation (ORA) [17]. Following SUMO that is developed by The Standard Upper Ontology working group [27], CORA is developed as top-level ontology for robotic and automation. It aims to provide clear definitions of the common concepts that will permeate all sub-ontologies to be developed within the ORA working group. This group is divided into four subgroups entitled Autonomous Robots, Service Robots, Industrial Robots and Upper Ontology/Methodology. CORA ontology encompasses a set of terms commonly used in Robotics and Automation, referring to ISO 8373 standard vocabulary. In order to enrich CORA further, CORAX is proposed to complement CORA with knowledge such as industrial design, physical environment, interaction, and artificial system that are not covered in SUMO [28]. An ontology model, named SRDL, is built to describe not only robots components but also their capabilities and actions [16]. The links among them allows the inferences about the ability of robots to perform certain actions. In summary, these ontological models provide unified and clear representation to robot or robotic systems, while most of them are upper-level models that cover the most common concepts suitable to all types of robots. In these semantic models, the terminologies related to the working process of robot are seldom involved. Although SUMO contains process concept, the specific information has not been defined for robot model CORA currently.

Industrial robot, as one of main branches of robot, its modeling is studied by many experts and engineers to describe manufacturing capability for supportive applications in industry, such as robot selection, knowledge reuse etc. The MC of IRs is achieved by actuators, sensors, computers and other supporting facilities and it is influenced by many factors such as cost, payload and processing interfaces [29]. Based on CORA, industrial assembly robot capability models are constructed consisting of robot, assembly action, end effector, assembly object, sensor, and result models [30]. The developed model is intended to be used for helping manufacturers to characterize the different capabilities their robots contribute to help the end user to select the appropriate robots for the appropriate tasks. In practical production, selecting an appropriate manufacturing resource is the guarantee of efficient production. A fuzzy multi-criteria evaluation algorithm is proposed to analyse the manufacturing capability of IRs from the perspective of technical and economic attributes [31]. These models provide the modeling of IRs, however they ignore the status of IRs during working process, without considering energy consumption. Under sustainable manufacturing circumstance, energy consumption of IRs should be taken into account as a respect of manufacturing capability of IRs.

Actually, in terms of energy consumption of IRs, many researches have already been carried out. However, most of them use mathematic method or dynamic kinematic analysis to achieve energy consumption model. An energy model of IRs is proposed based on the velocity and acceleration of motions [32].An approach where systematic mapping the energy consumption for movement between points in robot's workspace is developed by creating data sets of energy consumption for standard movement elements in [33]. The power consumption and dynamic behaviour analysis of industrial robot under several operation conditions are addressed in [18, 34]. An energy consumption model for each operations of multi-rotob system is explored to optimal production scheduling [19]. The Modelica-based simulation tool, CATIA Systems Dynamic Behaviour Modeling (DBM) is used for creating a digital model of industrial robot to calculate and predict energy consumption. Moreover, some of researches establish energy model by making use of process information. By measuring method, IRs' energy is depicted in different trajectories, weight tools, workpiece position and movement speed [35]. An energy model for the collaborative operation of multiple robotics is built according to diverse trajectories [36]. An IR's energy model is constructed based on spare time and reset time during running cycle [37]. Although these models can obtain energy consumption of IRs, they are unable to offer semantic interpretation for the attributes and the relationships between those attributes. In cloud manufacturing, it is significant to describe energy consumption by semantic pattern to support intelligent IRs' applications.

3. Unified concept model of sustainable manufacturing capability (SMC) of IRs

3.1 The definition of IRs and SMC

According to the ISO standards, IR is an automatically controlled, reprogrammable, multi-purpose, manipulator programmable in three or more axes, which may be either fixed in place or mobile for use in industrial automation applications [38]. It usually has multiple degrees of freedom, from which some axis are used for position of tool centre point (TCP) and some are for the orientation. It automatically executes tasks with its own power and control mechanisms by either human instructions or pre-programmed procedures set by artificial intelligence techniques.

IRs can be categorized in terms of control mechanism, function and motion mode. According to the control mechanism, IRs can be categorized into point robots and continuous path robots. Point robots control an actuator from one point to another, which are generally used for spot welding, loading, unloading and handling. Continuous path robots make an actuator move at a given track, which are suitable for continuous welding and coating. In terms of functions, IRs can be divided into handling robots, painting robots, welding robots, palletizing robots and assembling robots [10]. Towards the arms' movement modes, it can be classified into rectangular robots, cylindrical robots, spherical robots and joint robots. In many cases, various types of IRs collaborate with each other in a workshop to complete a task, which makes it necessary to build a unified model to realize centralized management to improve efficiency.

The sustainable manufacturing capability of IRs expresses the manufacturing capacity of IRs in a view of sustainable development. Manufacturing capability of IRs is the ability to complete manufacturing task. Referring to current literature, sustainability is defined with three dimensions: environmental, social and economical and technology [39]. Sustainable manufacturing can be defined as the ability to smartly use natural resources for manufacturing, by creating products and solutions that are able to satisfy environmental, economic and social objectives, thus preserving the environment, while continuing to improve the quality of human life [39, 40]. During the use of IRs, energy consumption is a main factor that directly affects energy cost and CO2 emissions, which concern strictly environmental saving. Manufacturers desire the usage of IRs in an energy-efficient way to reduce cost and increase productivity. In our current progress, we mainly study the sustainable manufacturing capacity of IRs covering energy consumption information. Reducing energy consumption deals with the motion planning, IR operating parameters optimization and IR operation schedule. Therefore, the SMC of IRs indicates much more than the structure and function of IRs itself. It also includes dynamic behaviours information and running condition related to energy consumption during the production process. For the sake of formal and semantic representation modeling of the SMC of IRs, we propose a unified description framework to organize the concepts and relationships at first, as the following section. This framework is extensible for the representation model of IRs' sustainable capability in the future.

3.2 A Unified Description Framework for the SMC of IRs

According to the above interpretation, we organize the concepts of SMC of IRs in terms of function attribute, structures information, action and process condition. Function attributes indicate the nominal parameters that IR is inherent to own. Structure information denotes the components of robot, tool and auxiliary information that contribute to the native capability of IRs. Action means the motion of IRs during manufacturing process. Process condition includes the real-time technical parameters and task progress of IR, which show the current status of IRs during processing. Activities and process condition are related to energy consumption that influences sustainability of IRs utilization.

SMC_IR={function attribute, structure information, activities, process condition}

Since the above concept model is related various resources and devices (such as manufacturer, sensor and auxiliary, etc) in IRs system, we propose a unified description framework to integrate these concepts and relationships. The framework includes generic part and specific part as a whole, as shown in Figure 1. All the solid lines express subsumption relation, and dot lines express other relations between different entities.

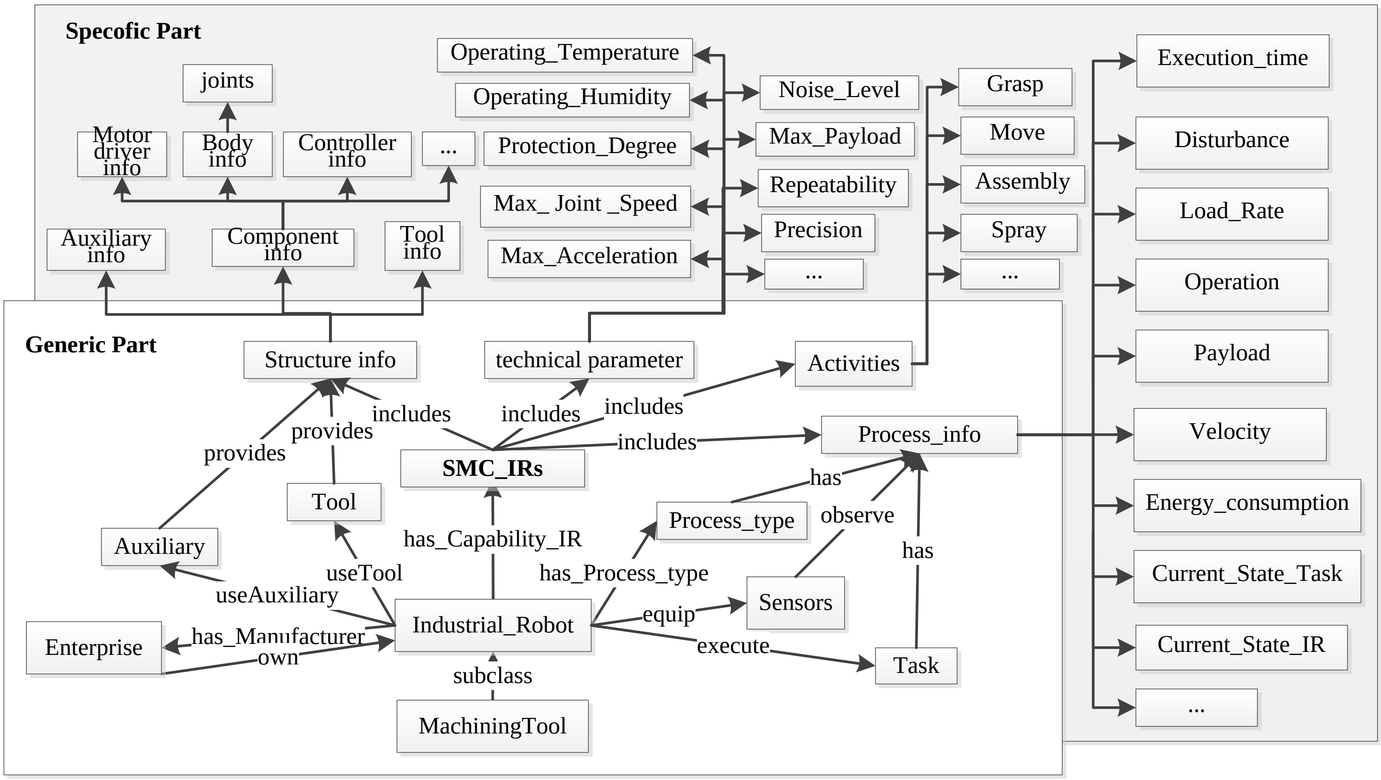


Figure 1. A unified description framework for the SMC of IRs

For generic part, common entities related to the SMC of IRs and the relationships between them are present. Besides Industrial Robot itself, the SMC also involves other entities, such manufacturer, tool, auxiliary, sensor, task and process type, which provide relevant information for the SMC of IRs. For example, tool indicates the end actuator of IRs in this case, including the gun, grasper and sucker.IRs achieves different functions by using corresponding tools. Through constructing these entities and their relations, generic part contributes to relational network of entities related to the SMC of IRs.

Specific part covers the concepts of SMC model for various types of IRs, including function attributes, structure information, activities and process condition. Among them, the first two types are constant and inherent for IRs. Activities show the change of state after carrying on motions that are possessed by different IRs. This status change usually involves the information of IR's structure and specific process. To process condition, it presents the real-time process information and task status, both of which are changeful during manufacturing process. Condition includes the real speed of IRs, payload, energy consumption and so on, which are quantifiable and updated at regular intervals. In the practical production, process information is acquired in real time by sensors that are installed on IRs. Obviously, the last two types of information are dynamic. Actually, this dynamic information is significant to express the SMC of IRs since it conduces to the calculation of environmental indicators. For example, the emissions of carbon and dust can be figured out by counting the energy consumption, which enables the model to judge whether the SMC of IRs is sufficient to do a task or not through inference rules. In the following, the condition information of IRs is addressed in detail.

3.3 Condition Information of IRs

Condition information perception facilitates the effective description of SMC of IRs during manufacturing process. The manufacturing process is reflected in real time through the IRs state, task state and energy consumption state, which are obtained by the information collection system set around IRs. According to the different types of collected information, sensors can be divided into temperature sensors, humidity sensors, audio sensors, RFID, cameras and so on. Each sensor has its own name, ID, type and manufacturer. As shown in Table 1, perceived attributes data are monitored by sensor and has value at specific time. For example, when the materials reach the workstation, the RFID readers placed on the workstation will read the RFID tags bound to the materials. Then the accurate arrival time of materials will be got, and the reasons that cause the delay of a task can be traced as well. In this way, it is possible to find out which material does not be supplied in time. Similarly, the cycle time of a task can be collected in the same way. These data are the basis to evaluate the production efficiency of IRs. In terms of the energy consumption, sensors such as meters are used to collect the energy consumption information of IRs under different operating conditions to support the classification and analysis of energy consumption.

Table 1. Properties related to sensory data

Subject	Predicate	Object	Notation
Process_info	observed_by	Sensor	Object property. It means that the property is monitored by a sensor.
Process_info	happen_at	--	Numerical property. It means that the monitoring activity is happened in a specific time.
Process_info	has_Value	--	Numerical property. It means that the property has a value at a time.

Drawing on existed manufacturing process model [41], the properties of *Process_info* are classified into the following three categories.

(1) Perceived properties. These properties are monitored by sensors in real time with quantifiable values which will be automatically updated at regular intervals. For example, *Payload*, *Velocity* and *Energy_Consumption* are all parts of perceived properties.

(2) Calculated properties. By analyzing and processing sensory data, the values of these properties can be figured out. For example, *Carbon_Emission* and *Toner_Dust_Emission* can be figured out through *Energy_Consumption*.

(3) Reasoned properties. This kind of properties represents implicit knowledge on IRs capability that is reasoned by axioms or SWRL rules in the unified model. The real-time speed, payload and loading state of IRs in the manufacturing process are the main factors that influence the energy consumption of IRs. Through defining these factors specifically, rules can be defined to infer the SMC of IRs for decision making of IRs selection. These will be demonstrated in Section 4.4 and Section 5.

Moreover, manufacturing process information of IRs must meet the needs of a task, such as duration limitation, required energy consumption etc. The properties of a task are listed in Table 2. Except *Should_Fulfill*, all the properties listed are numerical properties. *Should_Fulfill* is an object property, which connects task with the process condition of IRs. The capability state of IRs will be reasoned by comparing these properties to those properties representing the manufacturing process. The semantic rules are defined in section 4.

Table 2. Properties description for task demands

Attribute	Notation

has_Task_Name	the name of a task
has_Deadline	the deadline of a task
has_D_Quanlified_Rate	the lowest qualified rate that a task requires
has_EC_Per	the highest average energy consumption of products that a task requires
has_D_Quantity	the quantity that a task requires
has_Task_ID	the ID of a task
Should_Fulfill	the manufacturing process of IRs should meet the needs of a specific task

4. Dynamic semantic description for the SMC of IRs

In order to support the intelligent and efficient utilization of IRs in automation industry, it is significant to apply suitable representation formalism to the concept model of SMC of IRs. According the above concept model, the SMC of IRs involves not only native function, actions and structure attributes, but also process condition, which is changeful with time and environment. Although OWL, an ontology language, is able to give an unambiguous description to the fact of world, it only offers a static structure, taxonomy and constrains, lacking the representation of dynamic behaviours. Hence, the combination of OWL and DDL is adopted here to supply the formal representation of SMC of IRs.

4.1 The semantic combination of OWL and DDL

In order to supply comprehensive knowledge representation formalism to the SMC of IRs, the semantic logic of OWL and DDL are integrated to describe the static concepts and state of motions. OWL is a semantic language for defining and instantiating web ontologies. It provides ontology structure with meaning by the model theoretic semantics of SHOIN description logic. Although it satisfies the representation description about static application domain, OWL has expression limitation on dynamic behaviours. DDL is an integration of description logic and propositional dynamic logic (PDL). PDL, which is originally conceived as formalism for reasoning about the behaviour of iterative programs, has been proved to be useful as the basis of logics for actions and planning [20].Combining with modal operators of PDL, description logic can be extended to express the dynamic dimension of facts in the world. Referring to [20, 21, 42, 43], the constructors and corresponding semantic interpretations of hybrid language are listed as the Table3.

Table3. The logic constructors and semantic interpretations of OWL and DDL

Constructor		Syntax	Semantics
SHION	Class	C	$C^{\mathcal{I}} \subseteq \triangle^{\mathcal{I}}$
	Top	$\top$	$\triangle^{\mathcal{I}}$
	Bottom	$\bot$	$\emptyset$
	Role	R	$R^{\mathcal{I}} \subseteq \triangle^{\mathcal{I}} \times \triangle^{\mathcal{I}}$
	IntersectionOf	$C_1 \sqcap C_2$	$C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}}$

	UnionOf	$C_1 \sqcup C_2$	$C_1^{\mathcal{I}} \cup C_2^{\mathcal{I}}$
	ComplementOf	$\neg C$	$\triangle^{\mathcal{I}} \backslash C^{\mathcal{I}}$
	SomeValuesFrom	$\exists R.C$	$\{a \in \triangle^{\mathcal{I}} \mid \exists b,(a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\}$
	AllValuesFrom	$\forall R.C$	$\{a \in \triangle^{\mathcal{I}} \mid \exists b,(a,b) \in R^{\mathcal{I}} \rightarrow b \in C^{\mathcal{I}}\}$
	maxCardinality	$\leq nR$	$\{ a \in \triangle^{\mathcal{I}} \mid \lvert\{b \mid (a,b) \in R^{\mathcal{I}}\}\rvert \leq n\}$
	minCardinality	$\geq nR$	$\{ a \in \triangle^{\mathcal{I}} \mid \lvert\{b \mid (a,b) \in R^{\mathcal{I}}\}\rvert \geq n\}$
	subClassOf	$C_1 \sqsubseteq C_2$	$C_1^{\mathcal{I}} \sqsubseteq C_2^{\mathcal{I}}$
	equivalentClass	$C_1 \equiv C_2$	$C_1^{\mathcal{I}} \equiv C_2^{\mathcal{I}}$
	subPropertyOf	$R_1 \sqsubseteq R_2$	$R_1^{\mathcal{I}} \sqsubseteq R_2^{\mathcal{I}}$
	equivalentProperty	$R_1 \equiv R_2$	$R_1^{\mathcal{I}} \equiv R_2^{\mathcal{I}}$
	transitiveProperty	$R \in N_{R+}$	$R^{\mathcal{I}}=(R^{\mathcal{I}})^{+}$
	inverseOf	R^{-}	$\{(a,b) \in R^{\mathcal{I}} \mid (b,a) \in R^{\mathcal{I}}\}$
	disjointWith	$C_1 \sqcap C_2 = \bot$	$C_1^{\mathcal{I}} \sqcap C_2^{\mathcal{I}} = \emptyset$
	one of	$\{a_1\} \sqcup \ldots \sqcup \{a_n\}$	$\{a\}^{\mathcal{I}} \subseteq \triangle^{\mathcal{I}}, \lvert\{a\}^{\mathcal{I}}\rvert = 1$
PDL	Atomic Action	α	$\{\langle u,v \rangle \mid u,v \in W, u \rightarrow_{\alpha}^{\gamma} v\}$
	Sequence Action	$\alpha ; \beta$	$\{\langle u,v \rangle \mid u,v,\omega \in W, u \rightarrow_{\alpha}^{\gamma_1} \omega \wedge \omega \rightarrow_{\beta}^{\gamma_2} v\}$
	Choice Action	$\alpha \cup \beta$	$\{\langle u,v \rangle \mid u,v \in W, u \rightarrow_{\alpha}^{\gamma_1} v \vee u \rightarrow_{\beta}^{\gamma_2} v\}$
	Iteration Action	α^{*}	$\{\langle u,v \rangle \mid u,v \in W, u \rightarrow_{\alpha}^{\gamma_1} v \vee u \rightarrow_{\alpha;\alpha}^{\gamma_2} v \vee u \rightarrow_{\alpha;\alpha}^{\gamma_3} v \vee ...\}$
	Test Action	$\varphi?$	$\{\langle u,v \rangle \mid u \in W, u \models \varphi\}$

In Table3, for the semantics of SHION, The interpretation $\bullet^{\mathcal{I}}$ is a semantic interpretation. The non-empty $\triangle^{\mathcal{I}}$ is the domain of the interpretation. Every atomic concept C is assigned with a set$C^{\mathcal{I}} \subseteq \triangle^{\mathcal{I}}$and every atomic role R is expressed by a binary relation $R^{\mathcal{I}} \subseteq \triangle^{\mathcal{I}} \times \triangle^{\mathcal{I}}$. The rest of constructors are derived from the interpretations of atomic concept and role. For the PDL logic, an atomic action αis formulated as $\alpha(x_1, x_2,...,x_n) \equiv (P_{\alpha}, E_{\alpha})$, whereα is the name of the action, $x_1, x_2,...,x_n$ are the action variables, P_A is the preconditions set that expresses the state *u* before implementing atomic action, and E_A is the post conditions expresses the state *v* after the implementation. The semantic interpretation for atomic action α is that if existing an evaluation mapping γ satisfies the P_{α} and E_{α} of the atomic action α, the state *u* would lead to state *v* through executing α, denoted as $u \rightarrow_{\alpha}^{\gamma} v$. The others constructors are used to present complex actions that are composed of atomic action.

In description logic, the expressiveness and reasoning should be equilibrium state. Although the combination of various logics makes description logic more expressive, it also leads to the complexity of reasoning. The fact that is expressed by the logic syntax should be checked satisfiability. The logic syntax should ensure the decidability of reasoning. A tableau decision algorithm has been studied for the satisfiability-checking rules for combination of DL and PDL in terms of concepts, formulas and actions [44]. These rules enable the consistency check of the hybrid logic language.

4.2 Fundamental Ontology of the SMC of IRs

OWL Ontology offers expressive knowledge representation formalism to static facts of domain. According to the concept model in section 2, the fundamental ontology for SMC of IRs is built by utilizing OWL ontology. The formal description of IRs ontology is defined as the following.

Definition 1- *Industrial_Robot*

According to the definition of IRs and the concept model of SMC_IRs in the last section, Industrial_Robot can be described formally by logic description, as the following:

$$Industrial\_Robots \equiv Robot \sqcap \forall(\text{has\_manufacturing\_activities.manufacturing\_activities}) \sqcap (\exists has\_SMC\_IR.SMC\_IR$$

Where, SMC_IR is composed of four aspects, which is expressed as following:

$$SMC\_IRs \equiv (\exists has\_structure\_info.structure\_IRs) \sqcup (\exists has\_function\_attributes.function\_attribute) \sqcup (\exists has\_manufacturing\_actions.manufacturing\_actions) \sqcup (\exists has\_process\_condition.process\_condition)$$

Industrial_Robot ontology denotes the robots that have sustainable manufacturing capability are regarded as industrial robots, which have structure information, function attributes, manufacturing activities and process condition. According to the various manufacturing activities, industrial robot class can be classified into several subclasses, as shown on Figure 2. In this figure, dot line expresses object property, and solid line expresses subclass relationship.

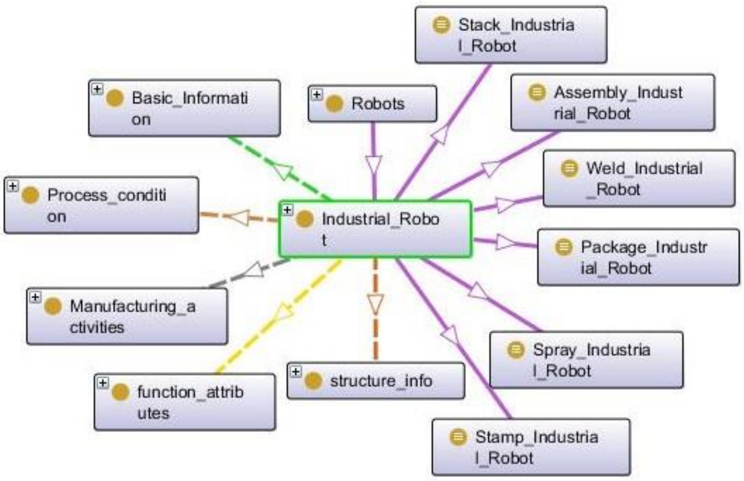


Figure 2. Ontology of *Industrial_Robot*

Definition 2- *Structure_IRs*

$$Structure\_IRs \equiv (\exists has\_tool\_info.tool\_IRs) \sqcup (\exists has\_component\_info.component\_IRs) \sqcup (\exists has\_auxiliary\_info.auxiliary\_IRs)$$

Structure_IRs is defined as the union of the information of the individuals of Tool_IRs, component_IRs and auxiliary_IRs. Tool_IRs include the tools that are specified to IRs for manufacturing activities. Component_IRs cover the native elements of IRs, such as motor, base and sensor etc. Auxiliary_IRs denote the devices that assist in machining.

Definition 3 - *Function_IRs*

$Function\_IRs \equiv (\exists has\_max\_acceleration.decimal) \sqcup (\exists has\_joint\_speed.decimal) \sqcup (\exists has\_max\_paload.integer) \sqcup (\exists has\_controlled\_axes.axes) \ldots$

Function_IR is defined by datatype properties that link individuals of concepts to data, which is used to express the native technical attributes of IRs. In OWL, there are plenty of built-in data types, such as string, decimal, integer etc.

Definition 4- *Process_condition*

$Process\_condition \equiv (\exists has\_state\_IR.string) \sqcup (\exists has\_current\_state\_task.string) \sqcup (\exists has\_cycle\_time.string) \sqcup (\exists has\_EC\_capability.string) \sqcup (\exists has\_load\_rate.decimal) \ldots$

Process_condition reflects the manufacturing process. It includes real-time status of an IRs and a task. Information such as qualified rate, current load and cycle time are also involved. In addition, availability, reliability and energy consumption which comprehensively express the SMC of IRs are contained as well.

Definition 5 - *Manufacturing_activities*

$Manufacturing\_activities \equiv \{Assembling\} \sqcup \{Packaging\} \sqcup \{Spraying\} \sqcup \{Stacking\} \sqcup \{Stamping\} \sqcup \{Welding\} \ldots$

Manufacturing_activities is defined as an enumerate class that is composed of different types of activities. These activities are disjoint with each other. Actually, these activities can be decomposed into atom actions. Every action reflects status transformation, which is expressed by DDL in the following sections.

4.3 Dynamic Description of Production Activities

As we mentioned in the last section, the production activities of IRs can be regarded as the combination of atomic actions. In order to give a formal description to these actions, we integrate OWL with DDL, which is a dynamic expansion of DL. With the extension of PDL in Table 3, seven atomic actions of IRs are defined in logic description pattern, shown on Table 4.

In Table 4, Action 1, 2 mean that the IRs is running and moving from position a to position b under the situation of holding product or not. Action1 express that when the end actuator of IRsholds nothing, the IRsmove from position a to position b. The result state is formulated as *located(r,b).* Action 2 shows thatwhen the end actuator of IRs holds a product, both IRs and the product move from position a to position b, namely *located(r,b)* and *located(p,b)*.

Action 3 means that the IRs is running and to catch a product. Before the IRs finishes the operation, the status of its end actuator is empty, expressed by is_Empty (r,true). After the IRs finishes the operation, the end actuator holdsthe product, and the status of actuator is not empty.

Action 4 means that the IRs is running and placing a product. Before the IRs finishes the operation, the status of its end actuator is holding a product. And after the IRs finishes the operation, the end actuator is empty.

Action 5, 6, 7 are respectively the assembling, welding and spraying of IRs.

Based on these atomic actions, complex actions are composed of them by action constructors (e.g. sequence and choice) in Table 3. So the complex actions of IRs are defined as follows.

Complex Action 8- Stacking

$$Stack \equiv Catch(r, p); Move(r, p); Place(r, p)$$

Stacking expresses the stacking of IRs. It is defined as an ordered sequence of three atomic actions, namely Catch, Move and Place.

Complex Action 9 – *Assembly_Update*

$$Assembly\_Update \equiv Catch(r, p); Move(r, p); Assembly(r, p)$$

Assembly_Update composes orderly three atomic actions: Catch, LoadMove and Assembly. This complex action is an update version of assembly to express complete assembly activity.Actually, manufacturing activities of IRs can be regarded as the compositions of atomic action and also complex actions. With the action operators, the activities of IRs can be expressed formally and logically.

Moreover, the constraints that an IRs should obey when performing production actions also can be formulated by DDL as the following:

$$\begin{aligned}&Stack;((Industrial\_Robot(r), Task\_Demand(t), Process\_IR(p),\\&should\_Fulfill(p,t) has\_Process\_IR(r,p), has\_Quanlity(p,a),\\&has\_D\_Quanlity(t,b), lessThan(a,b))?; Stack)^{*}; Equal(a,b)?\end{aligned}$$

It means that an IRs is stacking some products. If the number of products finished is less than the required number, the IRs will repeatedly execute the action until they're equivalent. Other constraints can be described like that.

4.4 The semantic description of energy consumption

Basically, the energy consumption of IRs is generated by the electrical machine, mechanical and auxiliary. Among these, the first two kinds consist of the main consumption of energy. Hence, we focus on the descriptions of these two types energy consumption. In practical manufacturing process, these two types energy consumption have close relationship with the process condition, like IRs speed, IRs payload and IRs actions. In order to describe the energy consumption, a state division method is proposed to label the energy consumption slices that are called interval energy consumption (iEC). We measure the iECs and label them by the corresponding state interval (iState).

In our semantic description, the state is defined by the intersections of speed, payload and behaviours type, shown as the following. And EC is labelled by the interval states.

$$iState \equiv State \sqcap (\exists has\_Activity.Activity) \sqcap (\exists has\_speed.decimal) \sqcap (\exists has\_payload.decimal)$$

$$iEC \equiv Run\_EC \sqcap (\exists has\_iState.iState)$$

For the practical manufacturing process, the time-based process can be split into state interval according to different operations. For most industrial robots, general manufacturing process can be expressed as Figure3. From starting the machine to shut up, the process can be divided into segments by operations and actions, such as start to move, action, replacement and so on. Every segment represents a state interval of IRs during manufacturing process. In Figure 3, the basic manufacturing process is divided into segmental states: standby, emptymove, catching, loadmove, placing, replacing. T1,T2,…T7 mark the beginning and ending of state segment. Actually, complex manufacturing process can be grouped sequentially by basic manufacturing processes with different actions, such as cutting, spraying etc.

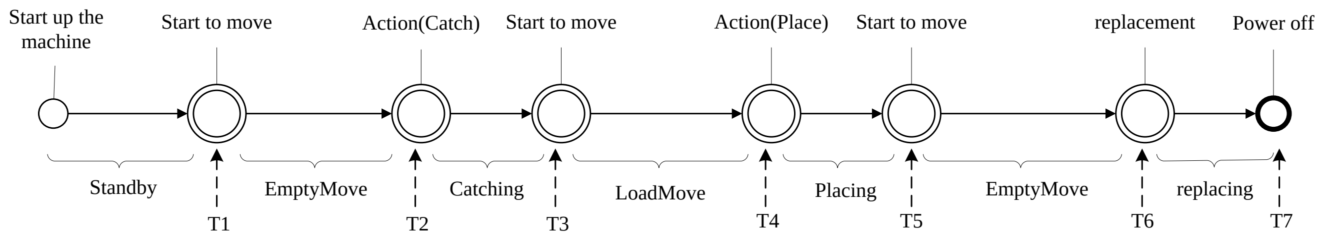


Figure3 a basic manufacturing process of IRs

Based on the above classification of states, the meta-states are described logically by the state definition, as the following:

1) $\mathrm{iState}_{Standby} \equiv State \sqcap (\exists has\_Activity.Power\_on) \sqcap (\exists has\_speed.0) \sqcap (\exists has\_payload.0)$

2) $iState_{EmptyMove} \equiv State \sqcap (\exists has\_Activity.EmptyMove) \sqcap (\exists has\_speed.decimal) \sqcap (\exists has\_payload.0)$

3) $iState_{Catching} \equiv State \sqcap (\exists has\_Activity.\ Catch) \sqcap (\exists has\_speed.decimal) \sqcap (\exists has\_payload.decimal)$

4) $iState_{LoadMove} \equiv State \sqcap (\exists has\_Activity.LoadMove) \sqcap (\exists has\_speed.decimal) \sqcap (\exists has\_payload.decimal)$

5) $iState_{Placing} \equiv State \sqcap (\exists has\_Activity.Plac)e \sqcap (\exists has\_speed.decimal) \sqcap (\exists has\_payload.decimal)$

6) $iState_{Replace} \equiv State \sqcap (\exists has\_Activity.Replace) \sqcap (\exists has\_speed.decimal) \sqcap (\exists has\_payload.decimal)$

According to these meta-states, meta-EC can be described as the following:

1) $iEC_{Standby} \equiv Run\_EC \sqcap (\exists has\_iState.iState_{Standby})$
2) $iEC_{EmptyMove} \equiv Run\_EC \sqcap (\exists has\_iState.iState_{EmptyMove})$
3) $iEC_{Catching} \equiv Run\_EC \sqcap (\exists has\_iState.iState_{Catcing})$
4) $iEC_{LoadMove} \equiv Run\_EC \sqcap (\exists has\_iState.iState_{LoadMove})$
5) $iEC_{Place} \equiv Run\_EC \sqcap (\exists has\_iState.iState_{Place})$
6) $iEC_{Replacing} \equiv Run\_\ EC \sqcap (\exists has\_iState.\ iState_{Replacing})$

Actually, since the end effectors that industrial robotic can be equipped are diverse, the actions and operations that robot executes are also various. There are more actions that can be described according to specific production applications. In practical manufacturing activities, process may contain single or multiple actions and operations. However, most processes can be regarded as the sequence combinations that are made up of movement and actions with different conditions. Hence, the proposed state division method is suitable to most industrial robotic, such as assembly robot, sorting and conveying robot, spraying robot etc.

The above partition method provides an upper-level description for energy consumption, which is a general division strategy. In fact, the interval state can be divided further for satisfy the demand of specific robotic application. For example, the movement interval can be cut further to describe joint's energy consumption, which benefits IRs trajectory planning and optimizing. In the current research, energy consumption description contributes to semantic reasoning for selecting IRs to satisfy energy demand of task. For more intelligent applications, the ontology model can be extended and enriched with further efforts in the future.

Table 4. The atomic actions definition of IRs

		Definition	
1	*EmptyMove*	*EmptyMove*(*a*,*b*,*r*) $\equiv(P_{EmptyMove}, E_{EmptyMove})$	$P_{EmptyMove}$={Position(a),Position(b),Industrial_Robot(r),has_Current_State(r,Running),located(r,a), has_Target_Position(r,b), is_Empty(r,ture)} $E_{EmptyMove}$={located(r,b)}
2	*LoadMove*	*LoadMove*(*a*,*b*,*r*,*p*) $\equiv(P_{LoadMove}, E_{LoadMove})$	$P_{LoadMove}$={Position(a),Position(b),Industrial_Robot(r),has_Current_State(r,Running),located(r,a), has_Target_Position(r,b), is_Empty(r,false),hold(r,p),Product(p)} $E_{LoadMove}$={located(r,b),located(p,b)}
3	*Catch*	*Catch*(*r*,*p*) $\equiv(P_{Catch}, E_{Catch})$	P_{catch}= {Industrial_Robot(r), has_Current_State(r, Running),Product(p), is_Empty(r,true)} E_{Catch}= {is_Empty(r,false), hold(r,p)}
4	*Place*	*Place(r,p)* $\equiv(P_{Place}, E_{Place})$	P_{Place}={Industrial_Robot(r),has_current_State(r,Running),Product(p),is_Empty(r,false),hold(r,p)} E_{Place}={is_Empty(r,true),¬hold(r,p)}
5	*Assembly*	*Assembly*(*r*,*p*,*base*) $\equiv(P_{Assembly}, E_{Assembly})$	$P_{Assembly}$= {Industrial_Robot(r),Product(p), Product(base),hold(r,p)} $E_{Assembly}$= {is_Empty(r,true),¬hold(r,p),is_Part_of(p,base)}
6	*Welding*	*Welding*(*r*,*p1*,*p2*) $\equiv(P_{Welding}, E_{Welding})$	P_{weld}={Industrial_Robot(r), has_Welding_Target(r,p1), has_Welding_Target(r,p2), Product(p1),Product(p2)} E_{Weld}={welded_with(p1,p2)}
7	Spraying	*Spraying*(*r*,*p*,*d*,*c*)$\equiv(P_{Spraying}, E_{Spraying})$	$P_{spraying}$={Industrial_Robot(r),has_Spraying_Target(r,p),Product(p),Spray_Gun(d), use_Tool(r,d),has_color(d,c),color(c)} $E_{spraying}$={has_color(p,c)}

5. Rule definition for sustainable manufacturing capability of IRs

Based on the above logic description of concepts, the SMC of IRs can be reasoned by defining rule in SWRL format. SWRL is a semantic rule language that supports OWL ontology. In SWRL, there are plenty of built-in operators that can make judgements for the reasoning, such as greaterThan, lessThan, etc.

By defining rules, the stability, energy consumption and production capability of IRs can be assessed against to the task demands that are described in section 3.3.

- Stability_Capability

Rule1:

Industrial_Robot(?r),has_Process_condition(?r,?p),has_Availability(?p,?a),greaterThanOrEqual(?a,0.8) ->has_stability_capability(?r, "Sufficient")

Rule2:

Industrial_Robot(?r),has_Process_condition(?r,?p),has_Availability(?p,?a),lessThan(?a,0.8) ->has_stability_capability(?r, "Insufficient")

Rule 1, 2 mean that if the availability of IRs during the manufacturing process is greater than or equal to 0.8, the stability of IRs is adequate, otherwise it's insufficient. The threshold 0.8 is given by experts, and can be appropriately adjusted according to specific circumstances. The availability is related to the failures in the manufacturing process. It's determined by MTTR (Mean Time ToRepair) and MTBF (Mean Time Between Failure).

- EC_Capability

Rule3:

Industrial_Robot(?r),has_Process_condition(?r,?p),Process_condition(?p),has_EC_Per(?p,?a),Task(?t), Task_has_demand(?t,?td)Task_demand(?td), has_D_EC_per(?td,?b),lessThanOrEqual(?a,?b) ->has_EC_Capability(?r, "sufficient")

Rule4:

Industrial_Robot(?r),has_Process_condition(?r,?p),Process_condition(?p),has_EC_Per(?p,?a),Task(?t), has_D_EC_per(?t,?b),greatThan(?a,?b) ->has_EC_Capability(?r, "insufficient")

Rule 3,4 mean that if the total energy consumption of process that IR is used to complete the task is less than or equal to the value that the task requires, the energy consumption capability of IRs is adequate, otherwise it's insufficient.

- Production_Capability

Rule5:

Industrial_Robot(?r), has_Process_condition(?r, ?p),Process_condition(?p), has_Current_State_Task(?p, *Done),has_Quantity(?p,?a),has_Qualified_Rate(?p,?b),has_Finished_Time(?p,?c),should_Fulfill(?p,?t), Task(?t),has_D_Qualified_Rate(?t, ?b1),has_D_Quantity(?t,?a1),has_Deadline(?t,?c1),greaterThanOrEqual(?a,?a1),greaterThanOrEqual(?b,?b1),lessThanOrEqual(?c,?c1)->has_Production_Capability(?b, "Sufficient")*

Rule 5 means that supposing a task having been done before the deadline, if the quantity of products is completed and the qualified rate are greater than or equal to the required values, and then the production capability of IRs is regarded as adequate, otherwise insufficient. The rule that declares the production capability of IRs is inadequate is not elaborated here due to space limitation.

6. Case Study and Implementation

6.1 The architecture of cloud-based manufacturing service platform for IRs

A cloud manufacturing service platform for IRs is developed, which fully demonstrates effectiveness and feasibility of the proposed modeling approach for the SMC of IRs. Through the network and cloud service platform, factories can access to manufacturing resources and capability services during the product lifecycle. These services are readily available, on-demand and reliable. The representation of IRs and its capability is achieved through an ontology model built by Protégé. Based on the model, SWRL rules are transformed into Jena rules, in order to analyse and reason by Jena plugin. The flow chart of the system is shown in Figure 4. There are four major services, namely, basic information service, equipment status service, task progress service and capability diagnosis service.

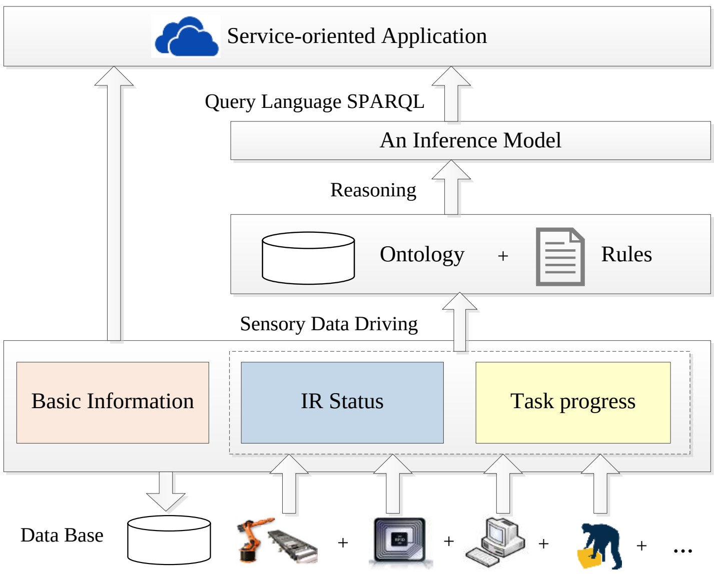


Figure 4. Function flow chart

(1) Basic information service. This module includes the basic information and technical parameters of IRs, which belong to static information that can be stored either in the ontology or the database. In this paper, this kind of information is uniformly stored in the database in order to obtain and display them more conveniently.

(2) Equipment status service. The state of IRs is dynamic during the manufacturing process. This article focuses on the SMC of IRs, namely the energy consumption and efficiency. This function

module displays the speed, load, track direction and energy consumption of IRs in real time. These data is monitored by sensors, and be updated once in a while.

(3) Task progress service. This module reflects the task progress, including the current state of task, working schedule, qualified rate and so on. Such information changing with the execution of task can be displayed dynamically.

(4) Capability diagnosis service. The capability of IRs is made of stability, energy consumption capability and production capability. After the task has been completed, the sensory data is written to the ontology model as instances' value by Jena plugin. These data serve as a basis for capability reasoning, including the actual finished time, completions, qualified rate, average energy consumption (the energy that per product consumed) and the stability of IRs. Jena rules and SWRL rules can be transformed each other. SWRL rules are reasoned in Protege development environment, and Jena can be implemented in platform framework environment.

The interface of the developed platform is shown in Figure 5. It can be found that IRs with different functions and structures used in the workshops are listed on the platform. Factories centrally manage the equipment through the platform according to the model proposed.

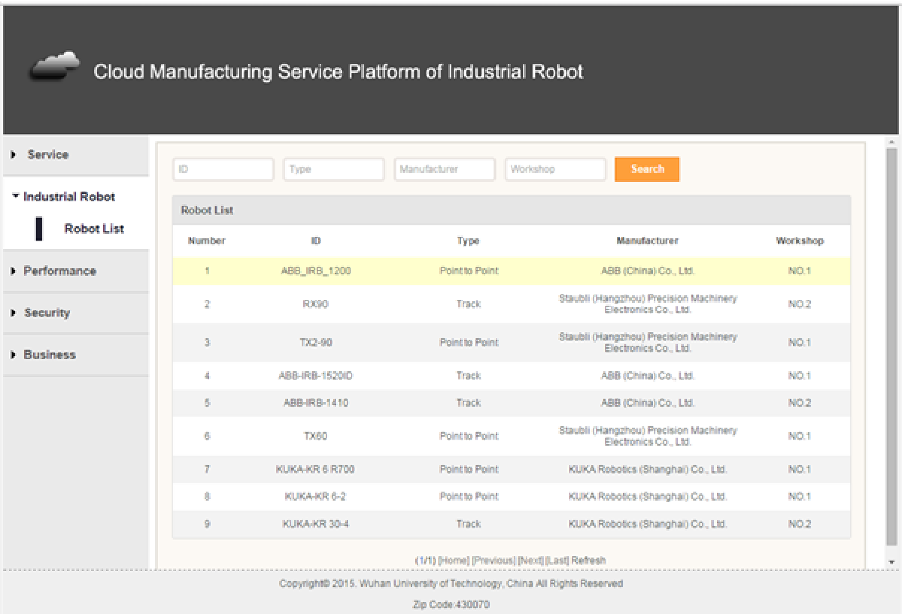


Figure 5. Cloud manufacturing service platform for IRs

6.2 Industrial Robot instance description

In order to verify the proposed ontology model, we take ABB_IRB_1200 as IRs instance, as shown in Figure 6. It is a flexible and multipurpose IRs manufactured by ABB Company. Comparing with other IRs, it has a small size narrowed by 15% while is faster about 10%. ABB_IRB_1200 not only has wide operating range, but also flexibility, rhythm and compactness which make it do well in transporting. The ABB_IRB_1200 is described in the proposed ontology model by inserting individuals, including technical parameters, motions, axis, and process state so on, as shown on Figure 7. The platform interface of the information is displayed on Figure 8. Jena, a Java API for ontology, provides a series of packaged function to handle ontology.

Figure 6. Robot ABB_IRB_1200

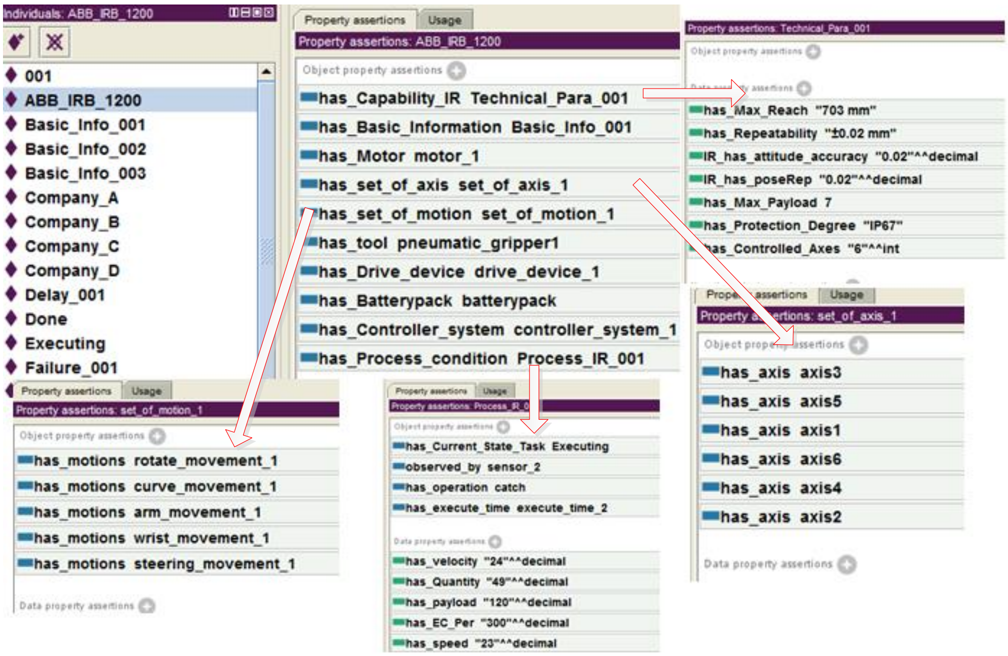


Figure 7. Individual description of IRs

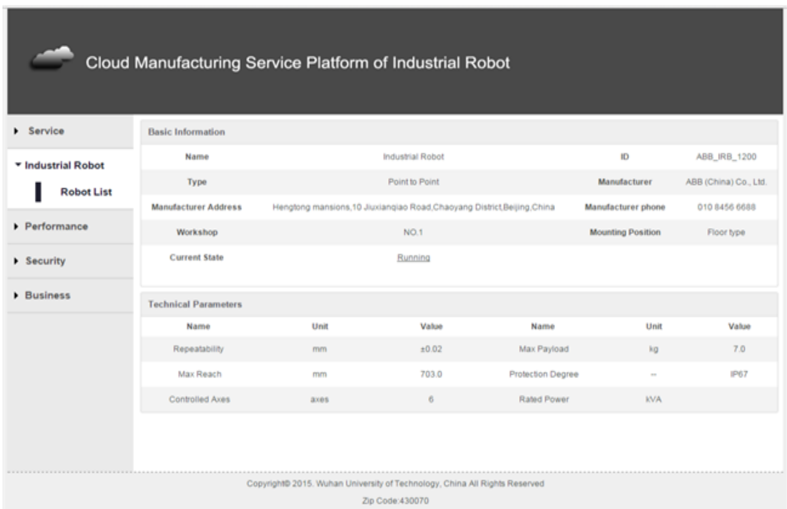


Figure 8. Visual interface of basic information and technical parameters

6.3 *Energy consumption measure and description*

For achieving energy consumption of the above IR during process, we equip ABB_IRB_1200 with intelligent electric meter to collect power data. The meter is a perception module designed based on ADE7878, the electric energy chip developed by Analog Devices Company. In our experiments, the four state intervals (EmptyMove, LoadMove, Catch, Place) are classified by speed and payload. During these four intervals, robot are operated in a speed range of 90~ 510 (mm/s) with four types payloads (55g, 110g, 165g, 220g). The values of collected power are shown on Figure 9. Every interval and power value can be inserted to ontology as instance. In Figure 10, the classes and instances of interval state and energy consumption are displayed. The total energy consumption of running process is the sum of all interval energy consumptions contained by the process. At last, the platform interface of energy consumption is shown on Figure 11.Besides energy consumption during IR running, it also includes the energy consumption in other situations, such as idle, ready, brake off etc.

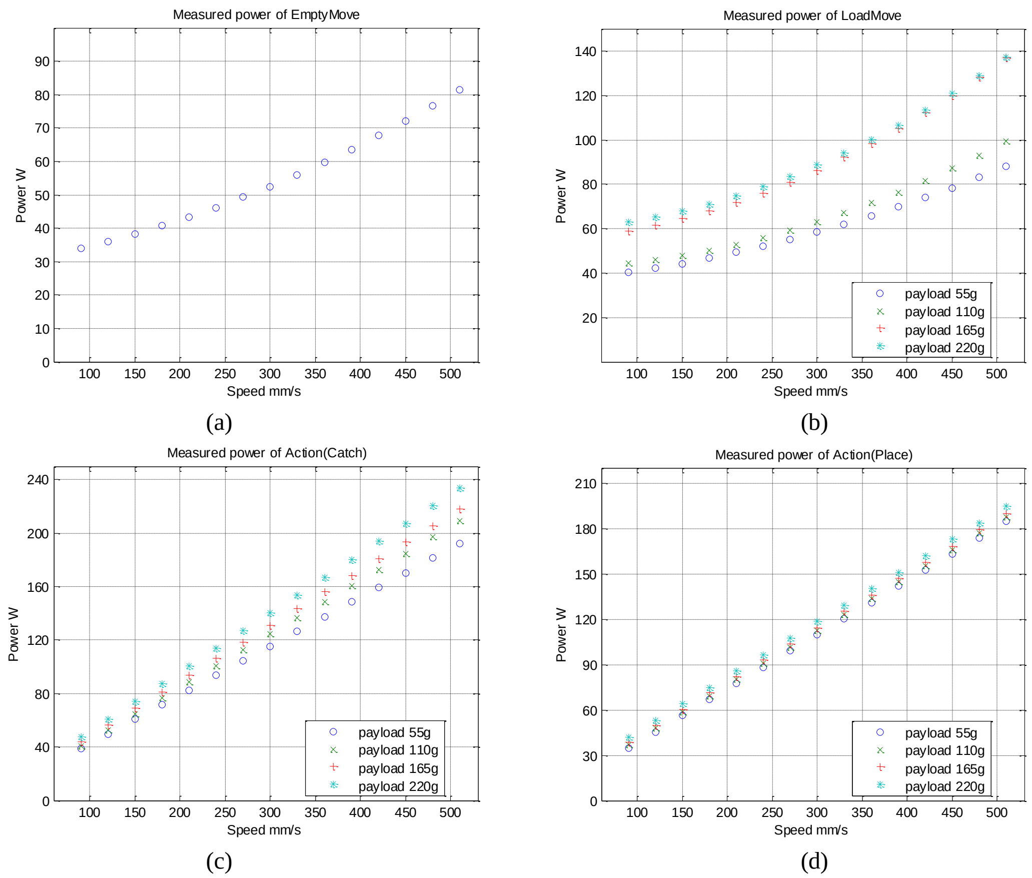
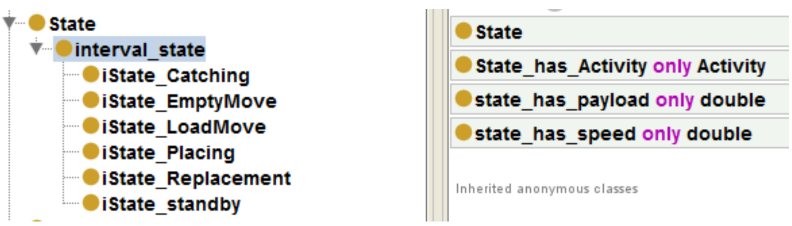
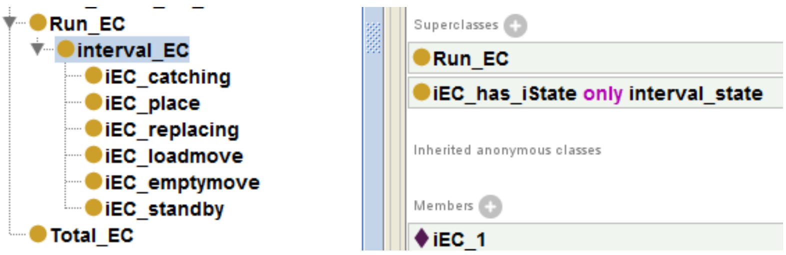


Figure 9. IR energy consumption in four process state intervals



(a) Interval state classes and discription



(b) Interval energy consumption classes and discription

(c) Individuals of iState and iEC

Figure 10. The classes and individuals of interval state and energy consumption

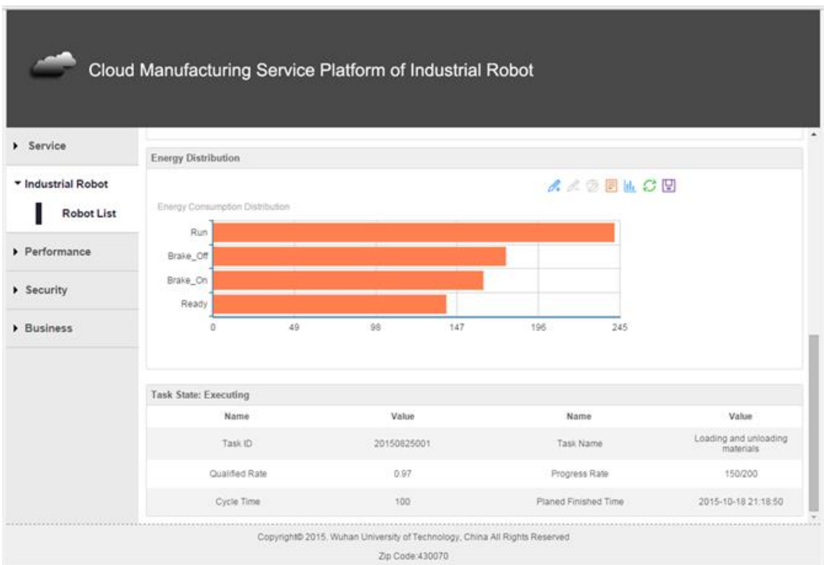


Figure 11. Visual interface of energy consumption distribution

6.4 Capability reasoning

In this section, the capability reasoning based on the proposed ontology model and rules are illustrated and verified with an elementary example. A task is supposed to process with the demands Task_Demand_1, including deadline, quantity, energy consumption, as shown on Figure 12. In this figure, the required quantity is 200, the latest finished time is 15, the max average energy consumption is 5 and the lowest qualified rate is 0.95. As can be seen from Figure 13, current process *Process_IR_001* of IR needs to meet the task requirement *Task_Demand_1*. When the task is completed, the qualified rate of IR is 0.97, IR's average energy consumption is 3.83, the product completed is 200, the finished time is 18 and the stability is 0.625.

```
<rdf:RDF xmlns="http://www.semanticweb.org/ontologies/2015/7/Ontology1440488637785.owl#"
   xml:base="http://www.semanticweb.org/ontologies/2015/7/Ontology1440488637785.owl"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
   xmlns:swrl="http://www.w3.org/2003/11/swrl#"
   xmlns:owl="http://www.w3.org/2002/07/owl#"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
   xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
   <!-- http://www.semanticweb.org/ontologies/2015/7/Ontology1440488637785.owl#Task_Demand_1 -->
   <owl:NamedIndividual rdf:about="http://www.semanticweb.org/ontologies/2015/7/ Ontology1440488637785.owl#Task_Demand_1">
         <rdf:type rdf:resource="http://www.semanticweb.org/ontologies/2015/7/Ontology1440488637785.owl#Task_Demand"/>
         <has_D_Qualified_Rate rdf:datatype="&xsd;double">0.95</has_D_Qualified_Rate>
         <has_Deadline rdf:datatype="&xsd;double">5.0</has_Deadline>
         <has_D_Quantity rdf:datatype="&xsd;int">200</has_D_Quantity>
         <has_D_EC_Per rdf:datatype="&xsd;double">5</has_D_EC_Per>
         <has_Task_Name>Loading and unloading materials</has_Task_Name>
         <has_Task_ID>20150825001</has_Task_ID>
   </owl:NamedIndividual>
</rdf:RDF>
```

Figure 12. Task demands described by OWL DL

```
<rdf:RDF xmlns="http://www.semanticweb.org/ontologies/2015/7/Ontology1440488637785.owl#"
   xml:base="http://www.semanticweb.org/ontologies/2015/7/Ontology1440488637785.owl"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
   xmlns:swrl="http://www.w3.org/2003/11/swrl#"
   xmlns:owl="http://www.w3.org/2002/07/owl#"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
   xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <!-- http://www.semanticweb.org/ontologies/2015/7/Ontology1440488637785.owl#Process_IR_001 -->
  <owl:NamedIndividual rdf:about="http://www.semanticweb.org/ontologies/2015/7/Ontology1440488637785.owl#Process_IR_001">
        <rdf:type rdf:resource="http://www.semanticweb.org/ontologies/2015/7/Ontology1440488637785.owl#Process_IR"/>
        <has_Availability rdf:datatype="&xsd;double">0.625</has_Availability>
        <has_Qualified_Rate rdf:datatype="&xsd;double">0.97</has_Qualified_Rate>
        <has_Finished_Time rdf:datatype="&xsd;double">6.0</has_Finished_Time>
        <has_Quantity rdf:datatype="&xsd;int">200</has_Quantity>
        <has_EC_Per rdf:datatype="&xsd;double">3.83</has_EC_Per>
        <should_Fulfill rdf:resource="http://www.semanticweb.org/ontologies/2015/7/Ontology1440488637785.owl#Task_Demand_1"/>
  </owl:NamedIndividual>
</rdf:RDF>
```

Figure 13. Dynamic parameters of manufacturingprocess described by OWL DL

Combining with the rules in section 5, the SMC of IRs will be reasoned and diagnosed. At first, the rules are transformed into Jena format. SPARQL is used to query the reasoned model. Figure 14 is the query statement. Figure 15 is the query result. The left picture is the reasoning in Protege, and the right one is the display on platform. It can be found that the energy consumption capability is sufficient, and although stability capability of IR is insufficient while the IR is still able to complete the task on time.

```
InfModel inf=model.getInfModel(ontModel);
String queryString1 = "PREFIX pre:<http://www.semanticweb.org/ontologies/2015/7/Ontology1440488637785.owl#> " +
      "SELECT ?Stability_Capability " +
      "WHERE {pre:IR_001 pre:has_Stability_Capability ?Stability_Capability}";
String queryString2 = "PREFIX pre:<http://www.semanticweb.org/ontologies/2015/7/Ontology1440488637785.owl#> " +
      "SELECT ?EC_Capability " +
      "WHERE {pre:IR_001 pre:has_EC_Capability ?EC_Capability}";
String queryString3 = "PREFIX pre:<http://www.semanticweb.org/ontologies/2015/7/Ontology1440488637785.owl#> " +
      "SELECT ?Production_Capability " +
      "WHERE {pre:IR_001 pre:has_Production_Capability ?Production_Capability}";
Query query=QueryFactory.create(queryString1);
QueryExecution qe=QueryExecutionFactory.create(query,inf);
ResultSet results=qe.execSelect();
……
```

Figure 14. SPARQL to query reasoned model

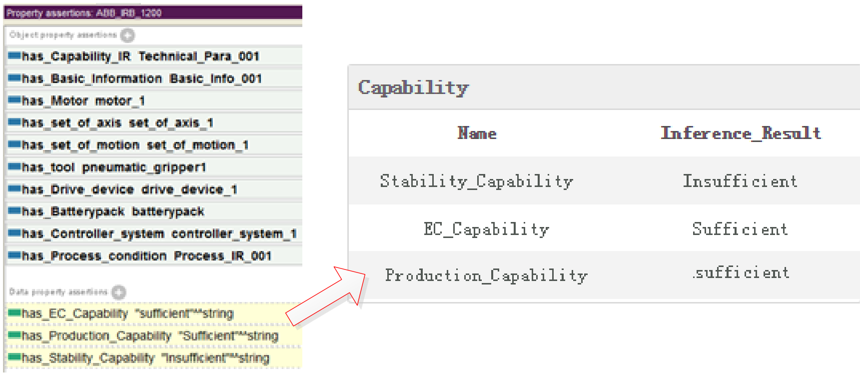


Figure 15. Reasoning results of capability

7. Conclusions

This paper presents dynamic and unified representation method for the SMC of IRs to support intelligent applications in cloud manufacturing. Through the unified description framework including generic part and specific part, the SMC of IRs is represented by organizing the relevant concepts and their relationships. It integrates these concepts from four aspects, including structure, function parameters, activities and process state, which satisfies the demand of effective usage and intelligent configuration of IRs. The proposed dynamic semantic description method contributes to the unambiguous and formal representation of SMC of IRs, which involves energy consumption and production activities of IRs during the manufacturing process. Based on the description format, three types of rules are defined to support the inference of IRs' sustainable capability. For the sake of practical application, a prototype system is developed to validate the feasibility of the proposed model. The system implementation shows that the developed description model successfully represents an IR instance and satisfies the sematic representation of energy consumption. Actually, the proposed ontology model is the base of IRs' capability knowledge base that is able to carry intelligent production configuration In future, the proposed model still needs to be extended according to the specific applications, and more factors should be considered to comprehensively describe the SMC of IRs, such as collaboration safety with human.

Acknowledgments

This is an extended version of a paper that was presented at the APMS 2015 International Conference Advances in Production Management Systems, Tokyo, Japan, September 5-9, 2015. This research is supported by National Natural Science Foundation of China (Grant Nos. 51305319 and 51675389), the International Science & Technology Cooperation Program, Hubei Technological Innovation Special Fund (Grant No. 2016AHB005), and Engineering and Physical Sciences Research Council (EPSRC), UK (Grant No. EP/N018524/1).

References

[1] Li B H, Zhang L, Wang S L, Tao F, Cao J W, Jiang X D, Song X and Cai X D. Cloud manufacturing: a new service-oriented networked manufacturing model [J]. Computer Integrated Manufacturing Systems, 2010, 16(1): 1-7.

[2] Xu X. From cloud computing to cloud manufacturing [J]. Robotics and computer-integrated manufacturing, 2012, 28(1): 75-86.

[3] Wang Vincent Xi, and Xun W. Xu, An interoperable solution for Cloud manufacturing [J]. Robotics and Computer-Integrated Manufacturing, 2013, 29(4): 232-247.

[4] Tao F, Zhang L, Guo H, Luo Y L and Ren L. Typical characteristics of cloud manufacturing and several key issues of cloud service composition [J]. Computer Integrated Manufacturing Systems, 2011, 17(3): 477-486.

[5] Ren L, Zhang L, Wang L and Chai F T X. Cloud manufacturing: key characteristics and applications [J]. International Journal of Computer Integrated Manufacturing, 2014: 1-15.

[6] Tao F, Feng Y, Zhang L and Liao T W. CLPS-GA: A case library and Pareto solution-based hybrid genetic algorithm for energy-aware cloud service scheduling [J]. Applied Soft Computing, 2014,19: 264-279.

[7] Wadhwa R. S., and Lien, T. K. Manufacturing Automation for Environmentally Sustainable Foundries. In Re-engineering Manufacturing for Sustainability, 2013: 171-174. Springer Singapore.

[8] Tao F, Bi L N, Zuo Y and Nee AYC. A hybrid group leader algorithm for green material selection with energy consideration in product design [J]. CIRP Annals-Manufacturing Technology, 2016, 65(1):9-12.

[9] Abele, E., Weigold, M., and Rothenbücher, S. Modeling and Identification of an Industrial Robot for Machining Applications. CIRP Annals - Manufacturing Technology, 2007, 56 (1): 387–390.

[10] Appleton, E., and Williams, D. Industrial robot applications. Springer Science & Business Media, 2012

[11] Chen Y H, and Dong F H. Robot machining: recent development and future research issues. The International Journal of Advanced Manufacturing Technology, 2013, 66(9-12): 1489-1497.

[12] Bugmann G, Siegel M, and Burcin R. A role for robotics in sustainable development?. IEEE (Institute of Electrical and Electronics Engineers) Africon, 2011: 13-15.

[13] Brossog M, Bornschlegl M, and Franke J. Reducing the energy consumption of industrial robots in manufacturing systems[J]. The International Journal of Advanced Manufacturing Technology, 2015, 78(5-8): 1315-1328.

[14] Meeussen W, Hsu J, Diankov R L, URDF - Unified Robot Description Format (April 2012), http://www.ros.org/wiki/urdf Accessed online 10/30/2016

[15] Xiao W L, Ji H and Dong S X. A STEP-compliant Industrial Robot Data Model for robot off-line programming systems [J]. Robotics and Computer-Integrated Manufacturing, 2014, 20(3): 114-123.

[16] Kunze, L, Roehm T, and Beetz M. Towards semantic robot description languages. Robotics and Automation (ICRA), 2011 IEEE International Conference on. IEEE, 2011.

[17] Prestes E, Carbonera J L, Fiorini S R, A.M.Jorge V, Mara A, Madhavan R, Locoro A, Goncalves P, Barreto M E, Habib M, Chibani A, Gerard S, Amirat Y and Schlenoff C. Towards a core ontology for robotics and automation[J]. Robotics and Autonomous Systems, 2013, 61(11): 1193-1204.

[18] Vergnano A, Lennartson B, Pellicciari M and Biller S. Modeling and optimization of energy consumption in cooperative multi-robot systems [J]. IEEE Transactions on Automation Science and Engineering, 2012, 9(2): 423-428.

[19] Matthias Brossog P, Kohl J, Merhof J, Spreng S and Franke J. Energy consumption and dynamic behavior analysis of a six-axis industrial robot in an assembly system[C]. Procedia CIRP23, 2014: 131-136.
[20] Wolter F and Zakharyaschev M. Dynamic Description Logics [J]. Advances in Modal Logic, 1998, 2: 431-446.
[21] Chang L, Shi Z, Gu T and Zhao L. A family of dynamic description logics for representing and reasoning about actions[J]. Journal of automated reasoning, 2012, 49(1): 1-52.
[22] [22]Mousavi S, Gagnol V, Bouzgarrou B C and Ray P. Dynamic modeling and stability prediction in robotic machining[J]. The International Journal of Advanced Manufacturing Technology, 2016: 1-13.
[23] Niu B, Zhang H. Model based control of industrial robot and implementation of its gain scheduling robust control[C]//Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on. IEEE, 2011: 2156-2162.
[24] [24]Estévez E, Sánchez-García A, Gámez-García J, Gomez-Ortega J and Satorres-Martinez S. A novel model-driven approach to support development cycle of robotic systems[J]. The International Journal of Advanced Manufacturing Technology, 2016, 82(1-4): 737-751.
[25] Steele J W, Wysk R A and Ferreira J C E. A resource-oriented tolerance representation scheme for the planning of robotic machine tending operations in automated manufacturing systems [J]. The International Journal of Advanced Manufacturing Technology, 2008, 38(7-8): 741-756.
[26] Vichare P, Nassehi A, Kumar S and Newman S. A Unified Manufacturing Resource Model for representing CNC machining systems[J]. Robotics and Computer-Integrated Manufacturing, 2009, 25(6): 999-1007.
[27] Niles I and Pease A. Towards a standard upper ontology[C].Proceedings of the international conference on Formal Ontology in Information Systems-Volume 2001. ACM, 2001: 2-9.
[28] Fiorini S R , Carbonera J L, Gonçalves P, A M Jorge V, Fortes Ray V, Haidegger T, Abel M, Redfield S A. Balakirsky S, Ragavan V, Li H, Schlenoff C and Prestes E. Extensions to the core ontology for robotics and automation[J]. Robotics and Computer-Integrated Manufacturing, 2015, 33: 3-11.
[29] Rao, R. V., Patel, B. K., and Parnichkun, M. Industrial Robot Selection using a Novel Decision Making Method Considering Objective and Subjective Preferences[J]. Robotics and Autonomous Systems, 2011,59 (6): 367-375.
[30] Kootbally Z, Kootbally Z. Industrial robot capability models for agile manufacturing[J]. Industrial Robot: An International Journal, 2016, 43(5): 481-494.
[31] Kahraman, C., Çevik, S., Ates, N. Y., and Gülbay, M. Fuzzy Multi-criteria Evaluation of Industrial Robotic Systems. Computers & Industrial Engineering, 2007, 52 (4): 414-433.
[32] Chemnitz, M., Schreck, G.,andKrüger, J. Analyzing Energy Consumption of Industrial Robots. 16th IEEE Conference in Emerging Technologies & Factory Automation (ETFA), Toulouse, 2011, September 1-4.
[33] Ystgaard P, Gjerstad T B, Lien T K and Nyen P A. Mapping energy consumption for industrial robots.19th CIRP International Conference on Life Cycle Engineering, Berkeley, 2012: 251-256.
[34] Matthias Brossog P, Bornschlegl M, Franke J. Reducing the energy consumption of industrial robots in manufacturing systems[J]. The International Journal of Advanced Manufacturing Technology, 2015, 78(5-8): 1315-1328.
[35] Rassolkin A, Hoimoja H and Teemets R. Energy saving possibilities in the industrial robot IRB 1600 control. Compatibility and Power Electronics. IEEE, 2011:226-229.
[36] Meike D, Pellicciari M and Berselli G. Energy Efficient Use of Multirobot Production Lines in the Automotive Industry: Detailed System Modeling and Optimization. IEEE Transactions on Automation Science & Engineering, 2014, 11(3):798-809.
[37] Pellicciari M, Berselli G, Leali F and Vergnano A. A method for reducing the energy consumption of pick-and-place industrial robots[J]. Mechatronics, 2013, 23(3):326–334.
[38] Standard I S O. 8373: 1994[J]. Manipulating Industrial Robots–Vocabulary,1994.
[39] Garetti M and Taisch M. Sustainable manufacturing: trends and research challenges[J]. Production Planning & Control, 2012, 23(2-3): 83-104.

[40] [40]Joung C B, Carrell J, Sarkar P and Feng S C. Categorization of indicators for sustainable manufacturing[J]. Ecological Indicators, 2013, 24: 148-157.
[41] Xu W J, Yu J J, Zhou Z D, Xie Y Q, Pham D T, and Ji C. Dynamic Modelling of Manufacturing Equipment Capability using Condition Information in Cloud Manufacturing" Journal of Manufacturing Science and Engineering 2015,137:(4).
[42] Li-min C L C. Action Theory Based on the Dynamic Description Logic DDL[J]. Computer Science, 2011,7: 045.
[43] Grosof B N, Horrocks I, Volz R and Decker S. Description logic programs: combining logic programs with description logic[C]. Proceedings of the 12th international conference on World Wide Web. ACM, 2003: 48-57.
[44] Chang L, Shi Z Z, Qiu L R and Lin F. A tableau decision algorithm for dynamic description logic[J]. Chinese Journal of Computers, 2008, 31(6): 896-909.