Int J Adv Manuf Technol
DOI 10.1007/s00170-014-5864-x

ORIGINAL ARTICLE

A knowledge representation for unit manufacturing processes

Yingzhong Zhang · Xiaofang Luo · Hong Zhang · John W. Sutherland

Received: 31 August 2013 / Accepted: 11 April 2014
© Springer-Verlag London 2014

Abstract A unit manufacturing process is a fundamental operation unit in the manufacturing process of a mechanical product. It requires extensive knowledge of various processes to select a suitable unit process in process planning and manufacturing services. In the face of a large number of unit processes, it is very necessary to research and develop the knowledge representation and information processing approach for unit processes. In this paper, the basic unit manufacturing process concepts and relationships between them are analyzed in detail. The semantic Web technology is introduced into the knowledge modeling for unit processes, and a new ontology-based knowledge representation model is presented. A formalization for taxonomic semantics and descriptions of manufacturing capability knowledge is presented. A rule base that consists of a set of rules using Semantic Web Rule Language (SWRL) is built, which is used to enable reasoning on taxonomic semantics and manufacturing knowledge of unit processes. A knowledge retrieval approach that integrates Web Ontology Language (OWL) ontologies and SWRL rules is proposed. Some knowledge retrieval examples employing Semantic Query-enhanced Web Rule Language (SQWRL) are given. A knowledge prototype system for unit processes is developed. This research is a new attempt to construct an open, scalable, and shared manufacturing process knowledge system.

Keywords Unit manufacturing process · Process knowledge · Knowledge representation · Ontology

Y. Zhang (✉) · X. Luo · H. Zhang
School of Mechanical Engineering, Dalian University of Technology, Dalian 116024, China
e-mail: zhangyz@dlut.edu.cn

J. W. Sutherland
Division of Environmental and Ecological Engineering, Purdue University, West Lafayette, IN 47907, USA

1 Introduction

The manufacturing process is a kind of human activities in which the raw materials, usually supplied in simple or shapeless forms, are manufactured into finished products with the specific shape, structure, and properties designed to fulfill specific requirements. The manufacturing process determines how a product is manufactured, so it is a crucial phase to meet the product design and quality requirements, and to reduce the manufacturing costs and environmental impacts.

The manufacturing process of a mechanical product usually consists of a series of sequenced individual operations. Each operation is known as a unit manufacturing process [1], such as casting, turning, and surface treatment. For brevity, in this paper they are referred to as unit processes. A unit process provides the fundamental manufacturing means and capabilities. In the manufacturing process planning and manufacturing services, the most concerned issue is the selections of unit processes, and then is to determine their sequences. It is required an in-depth understanding of the unit processes to select a suitable unit process, which involves a variety of unit process knowledge. For example, when selecting a unit process, what manufacturing capabilities that a unit process can provide, and what equipment, tools and other resources that a unit process relates to should be best understood. Therefore, it is necessary to research and establish a knowledge representation for unit processes.

The product manufacturing knowledge is a kind of special knowledge that illustrates how a product can be best produced. The knowledge representation and modeling for manufacturing processes have attracted much research interest in the areas of the computer-aided process planning (CAPP) and knowledge-based engineering over the past years [2]. Unfortunately, the progress of research in this area is small. Most of process knowledge still remains in the minds of experienced engineers, and in the engineering handbooks.

Published online: 11 May 2014

There are a large number of unit processes that have been commercially used in the manufacturing field. In addition, with the development of technologies, many advanced unit processes are constantly emerging. Furthermore, a wide variety of unit processes involves mechanical, thermal, chemical, and electronics. Therefore, it is difficult to give all unit processes a unified description and knowledge representation if only the traditional knowledge representational methodologies such as rule-based, case-based, frame networks and neural networks are employed. In fact, the knowledge of unit processes can be abstracted as a set of concepts, relationships, and constraints among them from complex unit processes, and the knowledge representation is to give them an explicit and formal specification.

Currently, the taxonomy or classification of unit processes is a typical conceptualized work for organizing and describing unit processes, which classifies them using some common characteristics. Many such classifications or taxonomies have been presented in the relevant technical literature, such as the DIN 8580 classification [3]. However, these taxonomies only provide a classification and hierarchical relationships for unit process terms according to their common characteristics. The meaning of a unit process represented by a term has not been explicitly defined, and furthermore the definition and modeling for semantic relationships between various objects within a unit process have been seldom involved.

Recently, the ontology and semantic Web technology has been widely applied into many research fields, such as gene ontology, and has much more advantages than other concept modeling technology in classification, sharing, and formalization. With this in mind, this paper analyzes the common characteristics, concepts, and the relationships between them in a unit process, and presents a new ontology-based knowledge representation methodology, which will lay the foundation for the knowledge-based CAPP system.

2 Literature review

2.1 Knowledge representation for manufacture processes

In general, the manufacturing activities are guided by process planning, which is a pivotal link between design and manufacturing. Process planning involves the selection of necessary manufacturing processes and the decision of their sequences to transform a designer's idea (namely the designed part) into a physical component economically and competitively. The selection and decision depend on the manufacturing knowledge heavily [4]. Therefore, since the CAPP concept was proposed, the manufacturing process knowledge representation has received much research effort in both academia and industry, and some research progresses have been made [2].

The generative process planning system had employed expert systems using the manufacturing process knowledge representation very early. In 1987, Hayes [5] proposed an expert system, Machinist, which can automatically make process plans for fabricating metal parts on a CNC machine tool. Chang [6] developed a process planning expert system that consists of feature refinement, process selections, tool selections, process sequencing, etc., in which the knowledge representation scheme used is frame based. Currently, with the development of information technology, the research on the process knowledge representation is developing toward much deeper level rather than a simple combination of existing models and an expert system.

A manufacturing process is a very complex activity, which involves a variety of knowledge, such as the product design and description, manufacturing operations, tools, machines, and the relationships between these entities. Rules are commonly used in the process planning knowledge representation because in various manufacturing engineering handbooks a great deal of knowledge can be represented as rule forms. The production rule is a widely used knowledge representation approach, which uses the basic form of IF–THEN constructs and can represent various kinds of complex statements. Liu and Wang [7] used a hybrid approach whereby knowledge-based rules and geometric reasoning rules are combined to sort out the sequence of interacting prismatic machining features.

Frame is also a general data structure, which is used to organize object knowledge. In general, the frame combines with the rule; the former is used to represent facts and the latter is used to represent know-how. In addition, the frame also combines with the semantic network. A semantic network is also a widely used knowledge representation approach, in which knowledge is represented as a graph with nodes corresponding to facts or concepts and arcs to relations between concepts.

In addition, a large number of process knowledge representation approaches and models have been proposed. Sormaz and Khoshnevis [4] presented an object-oriented knowledge representation scheme in which each process is represented by a frame with three groups of slots defining its attributes, methods and relations. Guerra-Zubiaga and Young [8] classified process knowledge into explicit, tacit, and implicit knowledge and designed a manufacturing knowledge model. Chen et al. [9] proposed a new process knowledge representation approach using a parameter flow chart which efficiently combines parameter information, flow chart technology, and visualization technology. Xu et al. [10] presented a novel process planning schema based on the process knowledge customization. Hao and Ma [11] proposed a knowledge-based auto-reasoning methodology in the hole-machining process planning.

2.2 Unit manufacturing processes

A unit process is the most fundamental part of the product manufacturing process. In the face of an extremely large number of existing and emerging unit processes, how to establish a unit process knowledge representation is very difficult. Currently, the published literature on unit process knowledge representation can be mainly divided into the following two categories:

1. Taxonomy of unit processes.

Taxonomy of unit processes is a primary knowledge representation work for organizing unit processes according to their some common characteristics. Many taxonomic approaches have been presented in the relevant technical literature. However, because there exist differences in understanding unit processes, the presented taxonomies are different. For example, DIN 8580 [3] classified the shaping level into original forming, transforming, separating, and joining, but Todd et al. [12] classified the shaping level into mass-reducing, mass-conserving, and joining. In addition, the meaning of a unit process represented by a term has not been explicitly defined, and the semantic relationships among various objects within a unit process have not been involved.

2. Unit process capabilities.

Unit process capabilities are referred to as the limitations or performance constraints and the range of applicability of a given unit manufacturing process. Typical process constraints are cutting speeds, feeds, cost, dimensional and tolerance limitations, surface finish, and shape that can be produced [13]. Process planning engineers usually use their knowledge and experience of process capabilities to select unit processes and tools. Currently, the knowledge representation of unit process capabilities is usually integrated into the CAPP system [13].

2.3 Knowledge representation for manufacturing processes based on ontology

An ontology is a formal explicit specification of a shared conceptualization [14]. From the definition of ontology, conceptualization ensures the concepts and relationships among concepts inside manufacturing processes to be defined in a systematic way. The formal explicit specification makes the knowledge representation be semantically shared, retrieved, and reused [15]. The semantic Web technologies based on ontology can effectively combine the domain knowledge with the information representation.

The Process Specification Language (PSL), initially proposed by the National Institute of Standards and Technology [16], is one of the early manufacturing ontologies. PSL theory is divided into three notions: language, model theory, and proof theory [17]. However, PSL mainly focus on the higher level knowledge representation for common manufacturing processes. In addition, PSL lacks supporting of formalization based on description logic.

Currently, the most advanced language for ontology representation is the Web Ontology Language (OWL) based on description logic [18]. OWL is recommended by the World Wide Web Consortium (W3C) as the ontology language of the semantic Web. The ontologies encoded with OWL have been widely applied into many research fields. Mikosa et al. [19] developed a system for distributed knowledge sharing and reuse in the Potential Failure Modes and Effects Analysis domain using an ontology based on description logic. Their results show that the proposed approach is adequate to support knowledge sharing and reuse. The published literature on the knowledge representation for unit manufacturing process is little. Some ontologies are aimed at providing a shared vocabulary for communication between machine control and process planning software applications. Garcia-Cresp et al. [20] presented a conceptual model based on ontology for the semantic representation of industrial manufacturing processes, but their works mainly focused on the process management. Lemaignan et al. [21] presented a proposal named MASON (Manufacturing's Semantics Ontology) for a manufacturing upper ontology, aimed to draft a common semantic net in the manufacturing domain using OWL. MASON is mainly applied to automatic cost estimation and semantic-aware multiagent system for manufacturing. Martínez-Pellitero et al. [22] proposed a new process-based ontology for the inspection process planning, but they mainly focused on the inspection planning activity.

3 Concept analysis for unit processes

3.1 An input/output model based on flows

Although it is difficult to give all unit processes a unified description due to their wide varieties, we can employ a general black box model based on input/output flows to represent a unit process, which is as shown in Fig. 1. As a unit process is only an operation unit in the product manufacturing chain, by means of the abstract of the actual unit processes the black model can be defined as

$$Unit_Process = \{Input_flows, Unit_operation, Output_flows\}$$

where *Input_flows* denotes the external substance and information flows which enter the unit process, *Output_flows* denotes the output substance flows which are converted from

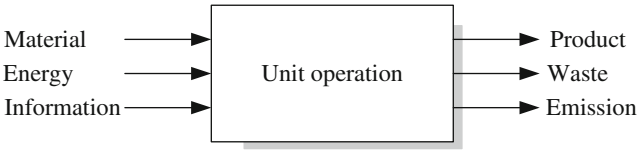


Fig. 1 A unit process input/output model based on flows

operating the input flows in the unit process, and *Unit_operation* denotes a specific unit operation that the unit process performs.

In general manufacturing processes, the input flows include material flows, energy flows, and information flows, and the output flows include product flows, wastes, and emission flows. The unit operation mainly includes shaping operations and modification material property operations. At the same time, there exist the following characteristics in the input/output flows of a unit process:

- Material flows.

Material flows are the major elements of a unit process. The input materials are usually rough parts or semi-finished parts, and the output materials are semi-finished or finished parts. Waste flows are a part of output materials. Material flows can be divided into the following three types as shown in Fig. 2.

(1) Through flows, corresponding to mass-conserving processes. In mass-conserving processes, the mass of the input material equals to the output material, which can be represented as $M_o=M_i$. The typical mass-conserving processes include the rolling process, the forging process, the bending process, etc.
(2) Converging flows, corresponding to assembly or joining processes. In the processes with converging flows, the multiple input materials are converged into an output material, which can be represented as $M_o=M_i^1+M_i^2$.
(3) Diverging flows, corresponding to mass-reducing processes. In the mass-reducing processes, the mass of the input material is reduced, which can be represented as $M_i=M_o^1+M_o^2$. Most of the machining processes have this flow type, such as the turning process and the drilling process.

- Energy flows.

Energy is the basic precondition for conducting a unit operation. During a unit operation, energy is converted into the power for the movement of workpieces and tools, and the work required for shaping the workpiece or the thermal energy for heating materials. The energy consumption is one of the important factors to evaluate a unit process.

- Information flows.

Information flows include shape-changing and property-changing information. Shape-changing information is converted into an interaction between the tool/die and the workpiece, and impressed on the workpiece's material so that the desired shapes can be obtained.

3.2 Concept analysis of a unit operation

A unit operation is the core of any unit manufacturing processes. Different unit processes are associated to the different unit operations which involve many different objects as shown in Fig. 3. We can abstract these objects as the following four concepts [1].

- Equipment: The equipment (e.g., machine tools) of a unit process converts the input energy into the movement of workpieces and tools, or the thermal energy to heat workpieces. Equipment determines the machining mechanism of a unit process and is the most important element affecting the efficiency, quality, and environmental performance of a unit process.
- Tools: Tools include cutting tools, grinding media, dies, molds, forms, patterns, electrodes, lasers, etc. A tool directly creates the shape of a workpiece, which is the main element affecting the workpiece's shape, dimensions, accuracy, and surface roughness.
- Workpieces: A workpiece is an object to be manufactured in a unit process. The material, structures, and size of a workpiece affects the selection of a unit process.
- Feature: A feature is an abstract geometry with a specific shape and attributes, which includes the design and manufacturing semantic information. In the manufacturing process, each shaping operation is usually used to relate to a specific feature of a workpiece. For example, a drilling process is used to machine a hole feature, and a turning process is used to shape cylindrical surfaces. Meanwhile, a feature is the tie to link the workpiece's shape, size, and accuracy with the processing methods, equipment, and tools.

4 A knowledge representation model for unit processes

4.1 Knowledge of unit processes

Knowledge can be considered as the theoretical or practical understanding of a subject, which generally includes facts, information, descriptions, or skills acquired through experiences or education. With the practice and development of manufacturing industries and technologies, a lot of unit process's domain knowledge has been created, which mainly includes the following three aspects:

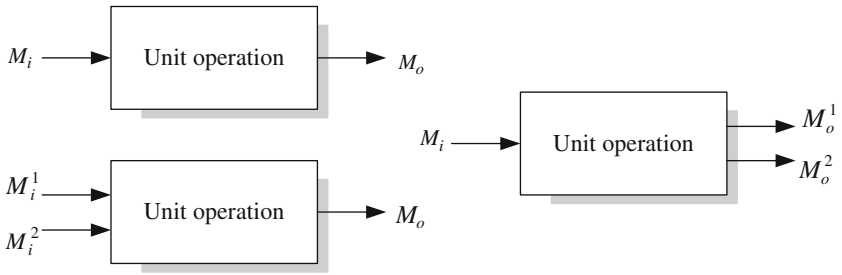


Fig. 2 Three types of material flows

1. Concepts of unit processes and their taxonomies.

A concept is an abstract form to represent entities and relationships. In the practice of manufacturing processes, many unit process concepts are created with the emergence of new manufacturing technologies and methods. For example, the milling process concept is represented by the term 'milling process' and implies rich contents or knowledge of the milling process, such as machine tools, cutting tools, shaping mechanism and methods, and constrain conditions.

In fact, the taxonomy of unit processes is an important organizational form of the unit process knowledge. However, different organizations or individuals may give different taxonomies according to their understandings. It is necessary to give an explicit specification for taxonomic semantics of the unit processes

2. Manufacturing capabilities of a unit process.

The manufacturing capabilities are the foundation for selecting a unit process in process planning, which are usually described by the shape, dimension, quality, and efficiency that the process can achieve [13]. The shape created on the workpiece surface is determined by the motion pattern among the equipment, tools and workpieces and the cutting pattern between the tool and the workpiece. Different unit processes have different manufacturing capabilities, even though in the same unit process there are big differences due to different combinations of the machines, tools, and workpieces. Therefore, the manufacturing capability is a kind of important manufacturing process knowledge.

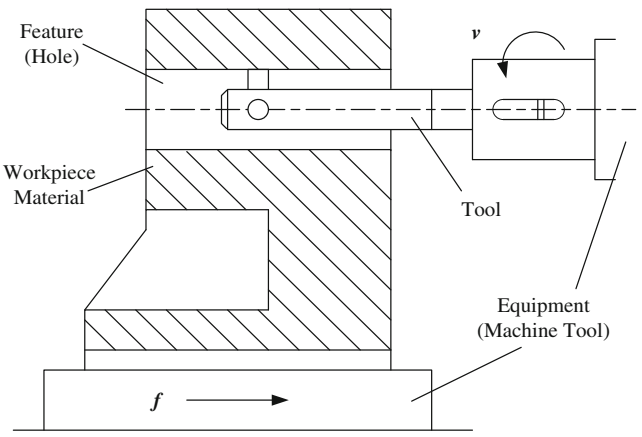


Fig. 3 Conceptual objects of a unit process

3. Process parameters of a unit process.

A process parameter is a numerical or other measurable factor forming one of a set that defines a system or sets the conditions of its operation, such as temperature, speed, and pressure, which determines the operating behavior of a unit process. The process parameters affect the manufacturing capabilities of a unit process directly, so they are a kind of know-how knowledge.

4.2 A knowledge representation framework for unit processes

A knowledge representation is a machine-interpretable representation of the real world, in which symbols serve as surrogates for real world domain artifacts, such as physical objects, events, and relationships, and can be conducted reasoning for the domain of interest [23]. In general, a knowledge-based system maintains a knowledge base which stores the symbols of the computational model in the form of statements about the domain, and it performs reasoning by manipulating these symbols.

As discussed in Section 2, the semantic Web technology based on ontology can effectively combine the domain knowledge with the information representation, and its formal and explicit specification can make the knowledge representation be semantically shared, retrieved, and reused. In addition, W3C also finalized Semantics Web Rule Language (SWRL) as an expressive OWL-based rule language [24]. SWRL allows users to write rules that can be expressed in terms of OWL concepts, so it provides more powerful deductive reasoning capabilities than OWL alone.

Based on the above analysis, a knowledge representation framework is proposed as shown in Fig. 4. This framework consists of three layers, namely the knowledge base layer, the reasoning layer, and the application interface layer.

The knowledge base layer, which is constructed on OWL DLs, includes a terminology box (TBox), an assertion box (ABox), and a rule base. The TBox defines the terminology, i.e., the concepts and relationships between concepts in the

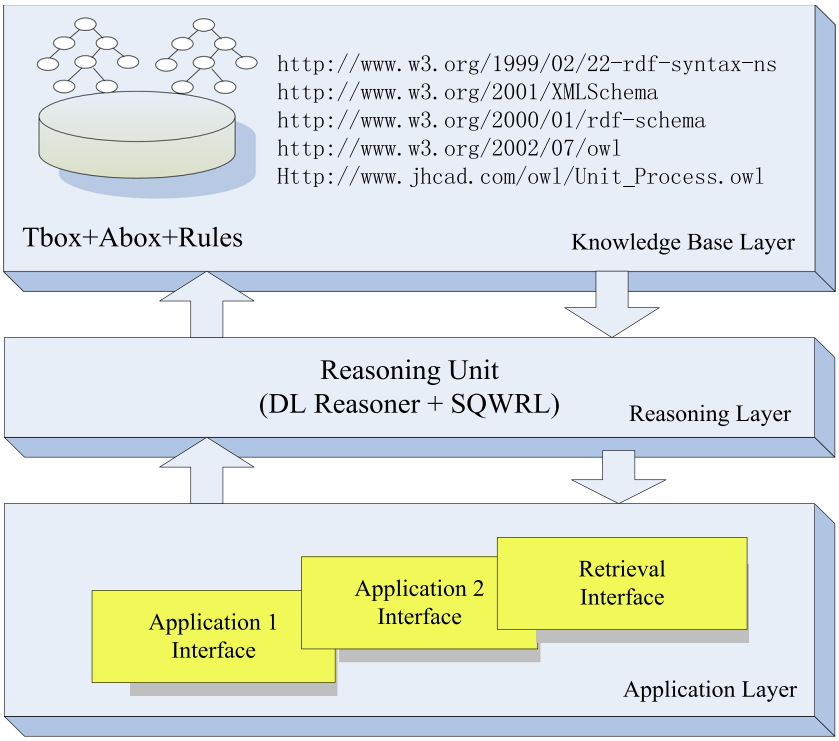


Fig. 4 The knowledge representation framework

manufacturing process domain. The ABox contains the instances in terms of concepts, for example, Drilling_H1 and Milling_F2 are the corresponding instances of the *Drilling* class and the *Milling* class in TBox respectively. Although OWL supports inference, it does not provide the reasoning rules such as "IF… THEN…", so its reasoning ability is very limited. Therefore, a rule base encoded with the SWRL language is defined in the knowledge base. A knowledge base can be represented as follows:

$$KB = \{TBox, ABox, Rules\}$$

The reasoning layer, which performs ontology reasoning tasks, includes two components. One is a Semantic Query-enhanced Web Rule Language (SQWRL) parser [25], which is used to convert the semantic retrieval requests from the application layer into SQWRL query sentences. Another is a reasoning engine. The reasoning layer generates specifications applied to the knowledge bases and returns the querying or inference results back to the application interface layer.

The application interface layer provides two interfaces: One is for human users, e.g., process planning engineers, to assign querying or reasoning tasks. Another is for querying or reasoning of application programs, such as the computer-aided process planning and the product life cycle analysis.

4.3 A knowledge representation model based on ontology

Based on the analysis and discussion in Section 3, and the conceptual elements of a unit process, this paper proposes a unit process ontology model as shown in Fig. 5. This ontology model constitutes the conceptual level to represent the knowledge of unit processes, in which the OWL ontology classes of concepts and the relationships between concepts are defined as follows.

4.3.1 Ontology classes for unit processes

The unit process ontology model shown in Fig. 5 includes nine top level concept classes, namely the *Unit_Process* class, *Inflow* class, *Outflow* class, *Equipment* class, *Tool* class, *Workpiece* class, *Feature* class, *Mfg_Capability* class, and *Proc_Parameter* class. The ontology definitions of the concepts and their taxonomies are given as follows:

1. *Unit_Process* class.

A unit process is an individual operation for the material blanks or semi-finished parts to be manufactured in order to obtain the shape, size, and material properties required by the product design. During the long-term industrial practice and technological development, a large number of manufacturing methods and means have been developed, and gradually formed a series of unit manufacturing process concepts, such as turning, milling, and grinding. These concepts are usually identified by process terminologies and implicitly express the information about the unit process's mechanism, means, conditions and tasks to be completed, etc. Although there exist many unit process taxonomies, most of them are classified

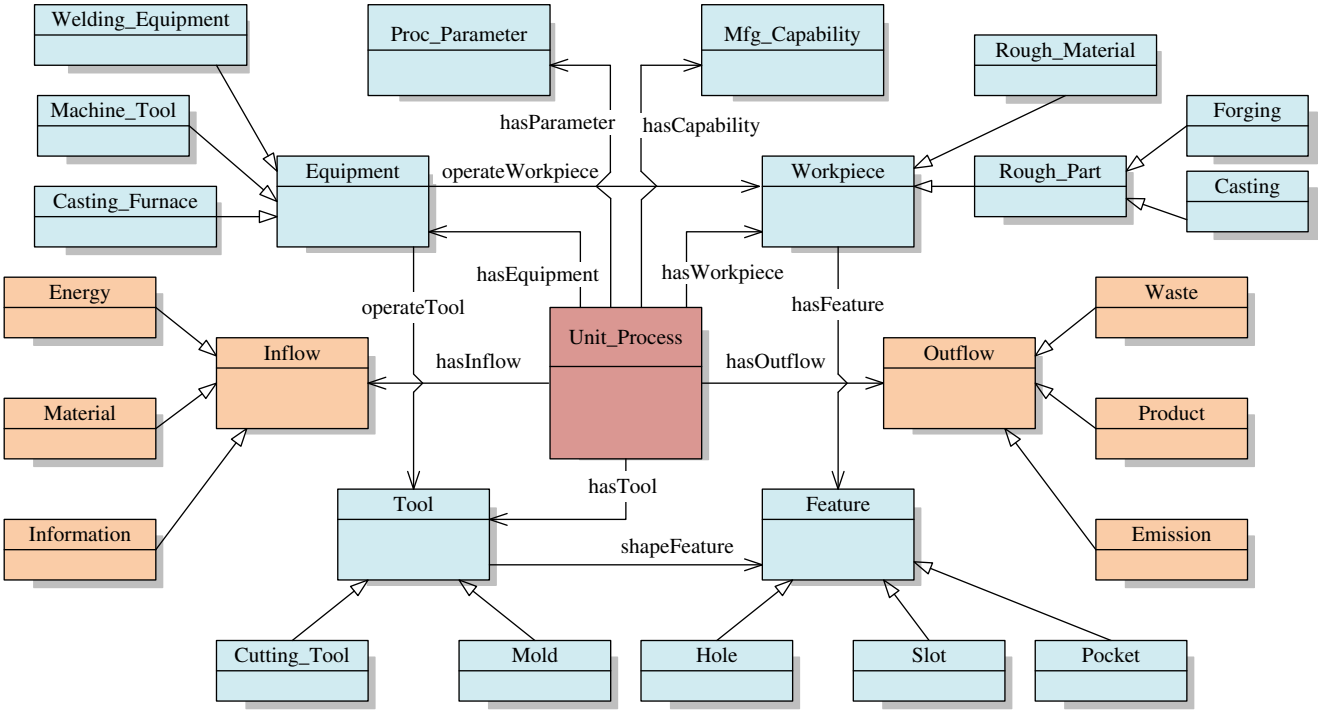


Fig. 5 An ontology model for unit processes

according to the physical process by which the configuration or structure of a material is changed.

According to two published taxonomies [3, 12] and the common characteristics of physical processes, this paper establishes a hierarchical ontology definition of the unit process concepts as shown in Fig. 6.

In Fig. 6, the *Unit_Process* class is defined as the base class of all unit process concepts (its parent class is "owl: Thing"), which derives two child classes: *Shaping_Process* class and *Non_Shaping_Process* class.

Shaping_Process class describes all concepts involved to unit processes by which a product's shape is changed, which derives the following three child classes according to the changes of the material's mass: (1) The *Mass_Reducing* class represents all unit processes during which the mass of the input material is reduced, such as separating processes and cutting processes. (2) The *Mass_Conserving* class represents all unit processes in which the mass of the output material is equal to the input, such as casting processes and deformation processes. (3) The *Joining class* represents all processes in which multiple parts or sub-assemblies are joined together.

Non_Shaping_Process class describes all concepts involved to unit processes by which the properties of the surfaces or bodies of the product material are improved, which derives the following two child classes according to the area where the properties of the parts are modified: (1) The *Material_Properties* class represents all the unit processes which alter the microstructure of a workpiece throughout its bulk in order to improve its mechanical properties, such as heat treatment, stiffen by deformation. (2) The *Surface_Finishing* class describes all the unit processes which alter the microstructure of a workpiece in a localized surface area and all the unit processes which coat a protection on the surface area of a workpiece.

2. *Inflow* class and *Outflow* class.

In the product manufacturing, the interactions between the unit processes, between the unit process and the environment are carried out by the input and output of substances. These substances are defined as the substance flows of unit processes and represented as the *Inflow* class and the *Outflow* class, respectively.

In the product life cycle activities, substance flows involve a variety of substances. As this paper only discusses the discrete manufacturing process, only the materials, energy, products, wastes, and emissions are defined in the flow ontology as shown in Fig. 7.

3. *Equipment* class and *Tool* class.

Equipment provides the movement, power, and control for workpieces and tools. Hence, equipment affects the efficiency, cost, and manufacturing quality of unit processes. In process planning, it is usually required to make a choice from the wide range of equipment. Therefore, the information about equipment is very important unit process knowledge.

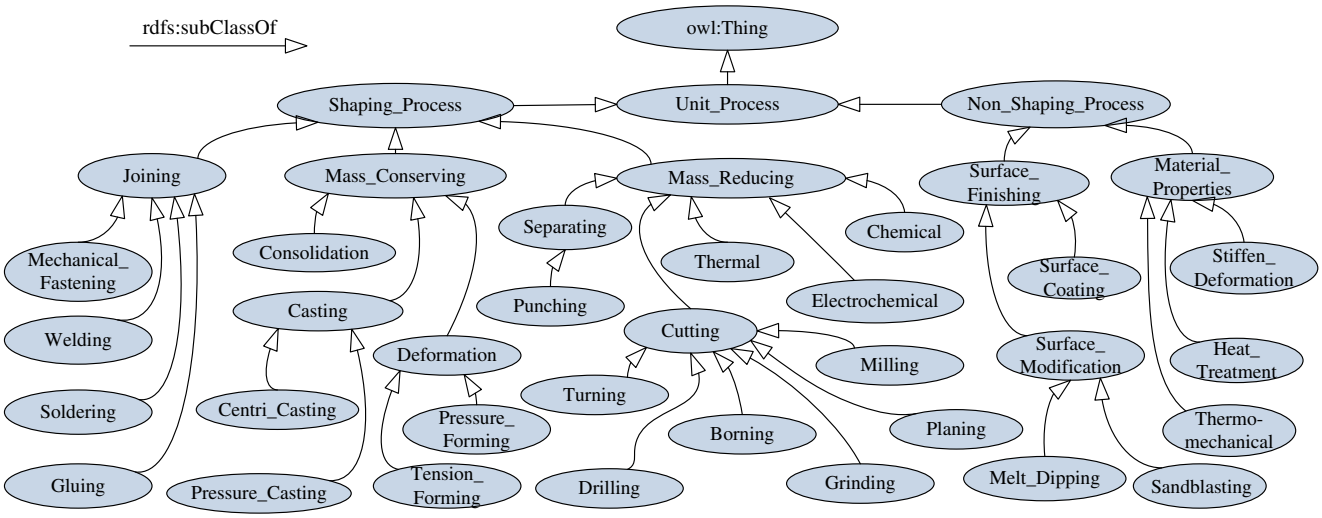


Fig. 6 Partial unit process ontology classes

Although there are a large number of equipment types related a unit process, we can use the *Equipment* class to represent the concept of equipment. The *Equipment* class is defined as shown in Fig. 8(a).

In a unit shaping process, the shape of a product is usually created by the relative motions between a tool and a workpiece, so the tool is an important concept element. As different unit processes employ different tools, it is necessary to classify the tools and construct the tool ontology. The *Tool* class is defined as shown in Fig. 8(b).

The shaping pattern created by the relative motions between a tool and a workpiece determines the shape that a unit process can produce on a workpiece. Hence, the shaping pattern is defined as a property of the *Tool* class.

4. *Workpiece* class and *Feature* class.

A workpiece is an input material of a unit process and also is the object to be processed. At the same time, in a unit process, a workpiece connects the input material flows to the output product flows. Before processing, the workpiece is the input flows of materials, and after processing, the workpiece is converted into the output product flows. A *Workpiece* class is defined to represent the concepts related workpieces.

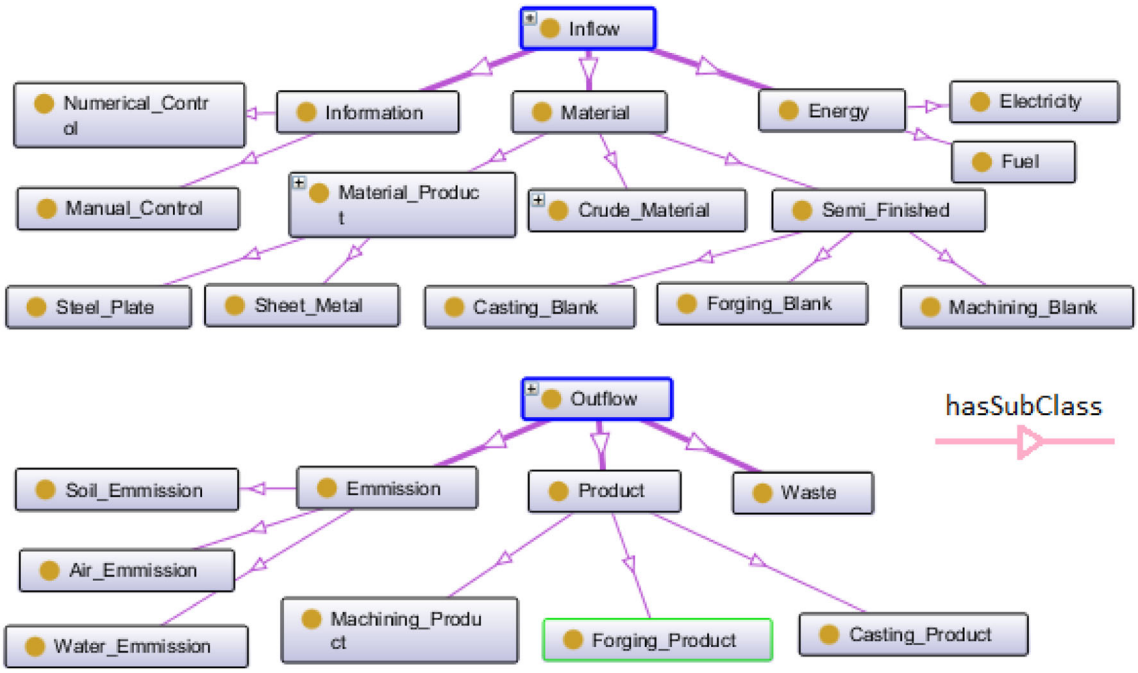
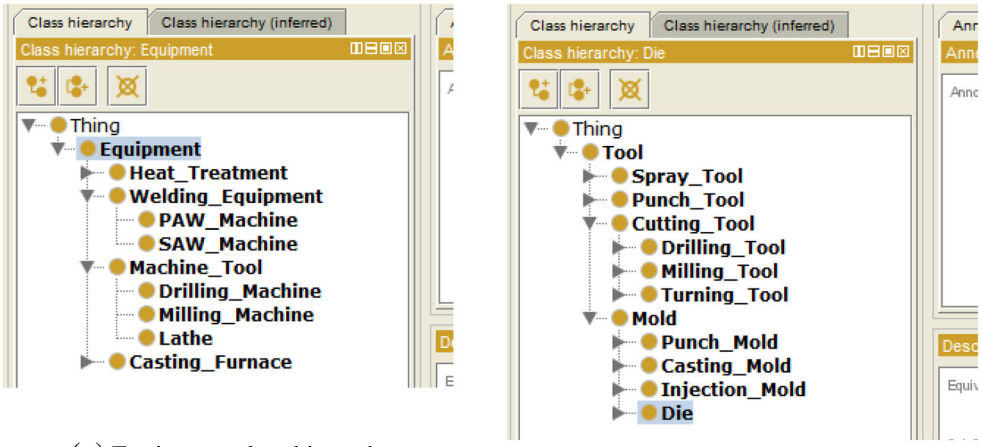


Fig. 7 The class hierarchy of the *Inflow* class and the *Outflow* class



(a) Equipment class hierarchy (b) Tool class hierarchy

Fig. 8 Partial definitions of the *Equipment* class and the *Tool* class

According to the material properties, structural forms, and manufacturing methods, the *Workpiece* class derives many other subclasses, such as the *Casting_material* class, the *Forging_material* class, and the *Sheet_part* class.

A feature is an entity with geometrical shape and engineering semantics. In general, shape of a part is described by features, so a *hasFeature* property is defined in the *Workpiece* class. In addition, the shape of a feature is the result of impressing tools or dies on a workpiece. There exist close relationships between the shape of a feature and the tool or mold/die. Therefore, a *shapFromFeature* property and a *shapToFeature* property are set in the *Tool* class, which is used to associate with the initial shape feature and the finished shape feature, respectively.

In computer-aided design and process planning, the feature technology has been widely used, and the taxonomy of the feature has been widely recognized. In this paper, a *Feature* class is defined and derives some commonly used subclasses such as the *Hole* class, the *Slot* class, the *Pocket* class, and the *Step* class.

5. *Mfg_Capability* class and *Proc_Parameter* class.

As discussed above, the manufacturing capability is an important part of the manufacturing process knowledge. In general, the manufacturing capability of a unit process is determined by its equipment, tools, workpiece, and machined feature, as well as its process parameters. It is required to give a flexible and scalable representation for manufacturing capabilities. In fact, the manufacturing capabilities are usually abstracted as a set of capability concepts and corresponding attribute values which can be compared and processed. Hence, we define an *Mfg_capability* class to represent the manufacturing capabilities of a unit process as shown in Fig. 9. However, different unit processes usually provide different manufacturing capabilities. For example, in the casting process the section thickness range, the melting temperature, etc. should be included into the capability description. Therefore, we define the *Mfg_Capability* class as the base class of the manufacturing capability, and all the specific capability concepts, such as the *Casting_Capability* and *Forging_Capability* class, are defined as its subclass in order to extend the manufacturing capability representation.

As the same as the method defining the manufacturing capability, we define a *Proc_Parameter* class to represent the process parameters of a unit process. The *Proc_Parameter* class can derive many subclasses to represent the process parameters of different types of unit processes. For example, the *Cutting_Parameter* subclass is used to represent the cutting speed, depth of cut, and feed rates in the machining process, and the *Casting_Parameter* subclass is used to represent the melting temperature, casting speeds, etc.

4.3.2 The relationships between the unit process concepts

The relationships between concepts of a unit process can be divided into the following four types:

1. The relationships between a unit process and its external substance flows.

We employ the *hasInflow* relation to associate a unit process with inflows, and the *hasOutflow* relation to associate a unit process with outflows..

2. The relationships between a unit process and its internal operating concepts.

We employ the *hasEquipment* relation to associate a unit process with equipment (e.g., machine tool, casting furnaces), the *hasWorkpiece* relation to associate a unit process with workpieces to be manufactured, and the *hasTool* relation to associate a unit process with tools.

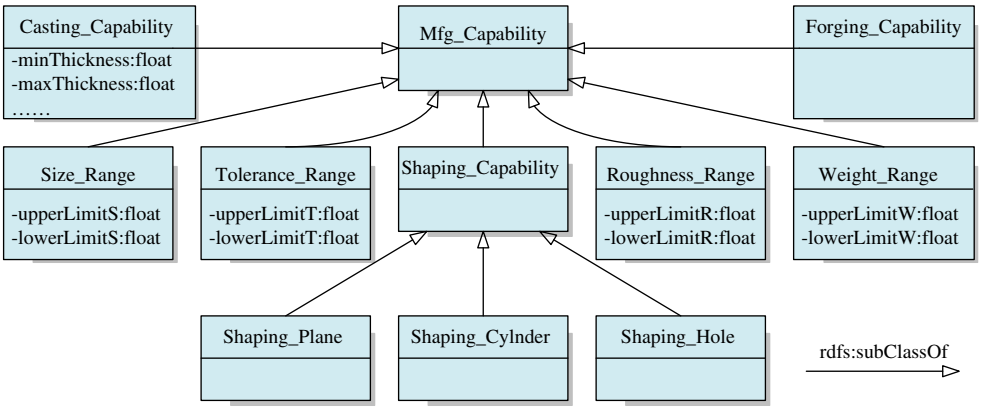


Fig. 9 Partial definitions of the *Mfg_Capability* class

3. The operating relations between the equipment and workpieces.

In general, there exist the following operating relations between the equipment and workpieces:

- Rotating workpieces. For example, in general lathe machining processes, the workpiece is usually rotated by the lathe machine tool.
- Moving workpieces. For example, in the machining processes of the planning machine or gantry milling machine, the workpiece is usually moved by the machine tools.
- Fixing workpieces. In most drilling processes and milling processes, the workpiece is fixed and does not move and rotate.
- Heating workpieces. In heat processes and heat treatment processes, the workpiece is heated.

Above relationships can be defined as a set of properties of OWL in a hierarchical form as shown in Fig. 10.

4. The operating relations between the equipment and tools.

Same as above, there exist the following operating relations between the equipment and tools:

- Rotating tools. A drilling machine or milling machine rotates the cutting tools.
- Moving tools. A turning machine, planar machine, or sawing machine moves tools.
- Fixing tools. The mold is usually a fixed tool.
- Heating Tools. In flame cutting, laser cutting, or welding processes, the tool is usually heated.

Above the operating relationships can be defined as properties of the OWL ontology in a hierarchical form as shown in Fig. 10.

5. The relations between a tool and manufacturing features.

The geometrical shape of manufacturing features is usually formed by the relative movement between the tools and workpiece. The *shapeFromFeature* property and *shapeToFeature* property of the *Tool* class are employed to represent the relationships.

4.3.3 Properties of a unit process ontology

In OWL ontology, the relationships between concepts are usually defined as properties of ontology classes. To sum up the defined properties, the main properties of the unit process ontology are given in Table 1.

4.4 Knowledge construction for the unit process

Above ontology model only provides a knowledge representation at the conceptual level. A unit process may employ different equipment, tools, and workpieces, etc. For example, a drilling process can employ a radial drilling or a vertical drilling, or even a lathe. Different combinations will result different manufacturing capabilities. We represent this combination with a set of instances of the concept classes in TBox, which constitutes the ABox of the unit process knowledge base. The ABox can be represented as:

$$\text{ABox} = \left\{P_1^1, P_1^2, \cdots, P_1^k, \cdots, P_m^1, P_m^2 \cdots, P_m^n\right\}$$

where m denotes that there are m unit process instances in the ABox, k denotes there are k combinations in the process instance P_1, and n denotes there are n combinations in the process instance P_m.

The knowledge base construction is a process of the knowledge capture, collection, and organization. This paper does not focus on the knowledge capture and only employs the method of the expert input. We have built more than 200 knowledge instances in the ABox of the knowledge base. Figure 11 illustrates partial instances of the drilling process and the casting process respectively. In Fig. 11, a rectangle node, an

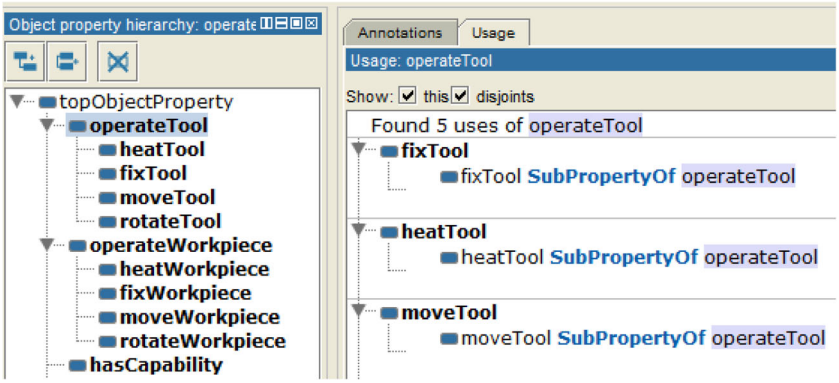


Fig. 10 The property definition of the operating relations

ellipse node, and a rounded rectangle node denotes an ontology concept, an ontology instance, and a value of the instance property, respectively.

5 Knowledge rules for unit processes

In the knowledge representation, there are many semantic relations that cannot be defined by the OWL ontology explicitly, such as causal relationships. Therefore, it is necessary to define new knowledge rules that can be added to the inference engine in order to meet the requirements for reasoning various relationships in the manufacturing process.

The relationships in ontologies can be represented by description logic based on predicate, which gives the relations precise semantics. In description logic, concepts are mapped to unary, and relations to binary predicates. The meaning of an element in a relation expression is determined by the domain and range defined in an ontology. For example, Unit_Process(x) denotes x is a unit process, and hasTool(x, y) denotes unit process x has tool y because the domain of the hasTool property is the *Unit_Process* class, and the range is the *Tool* class.

The knowledge rules for a unit process mainly include the rules for taxonomic semantics of unit process concepts, the shaping rules, and the process scheme rules.

5.1 Explicit rules for taxonomic semantics of unit processes

In Fig. 6, a semantic hierarchical structure for taxonomies of unit processes is given. However, there still lacks an explicit specification for taxonomic semantics of unit processes. The following gives a part of formal rule definitions with the predicate logic for the taxonomic semantics of unit processes, which can be used to label the unit process instances automatically.

5.1.1 Shaping processes and non-shaping processes

The common characteristic of a shaping process is to create shapes on the workpiece by shaping tools (e.g.,

Table 1 Partial defined properties of the unit process ontology

Property	Type	Description	Domain	Range
hasName	Data	Text description	Unit_Process	xsd:string
hasCapability	Object	To associate capabilities	Unit_Process	Mfg_Capability
hasParameter	Object	To associate process parameters	Unit_Process	Proc_Parameter
hasEquipment	Object	To associate an equipment	Unit_Process	Equipment
hasWorkpiece	Object	To associate a workpiece	Unit_Process	Workpiece
hasTool	Object	To associate a tool	Unit_Process	Tool
hasFeature	Object	To associate a feature	Workpiece	Feature
shapeToFeature	Object	To associate a finished feature	Tool	Feature
hasInflow	Object	To associate input flows	Unit_Process	Inflow
hasOutflow	Object	To associate output flows	Unit_Process	Outflow
operateWorkpiece	Object	Equipment operates workpieces	Equipment	Workpiece
operateTool	Object	Equipment operates tools	Equipment	Tool

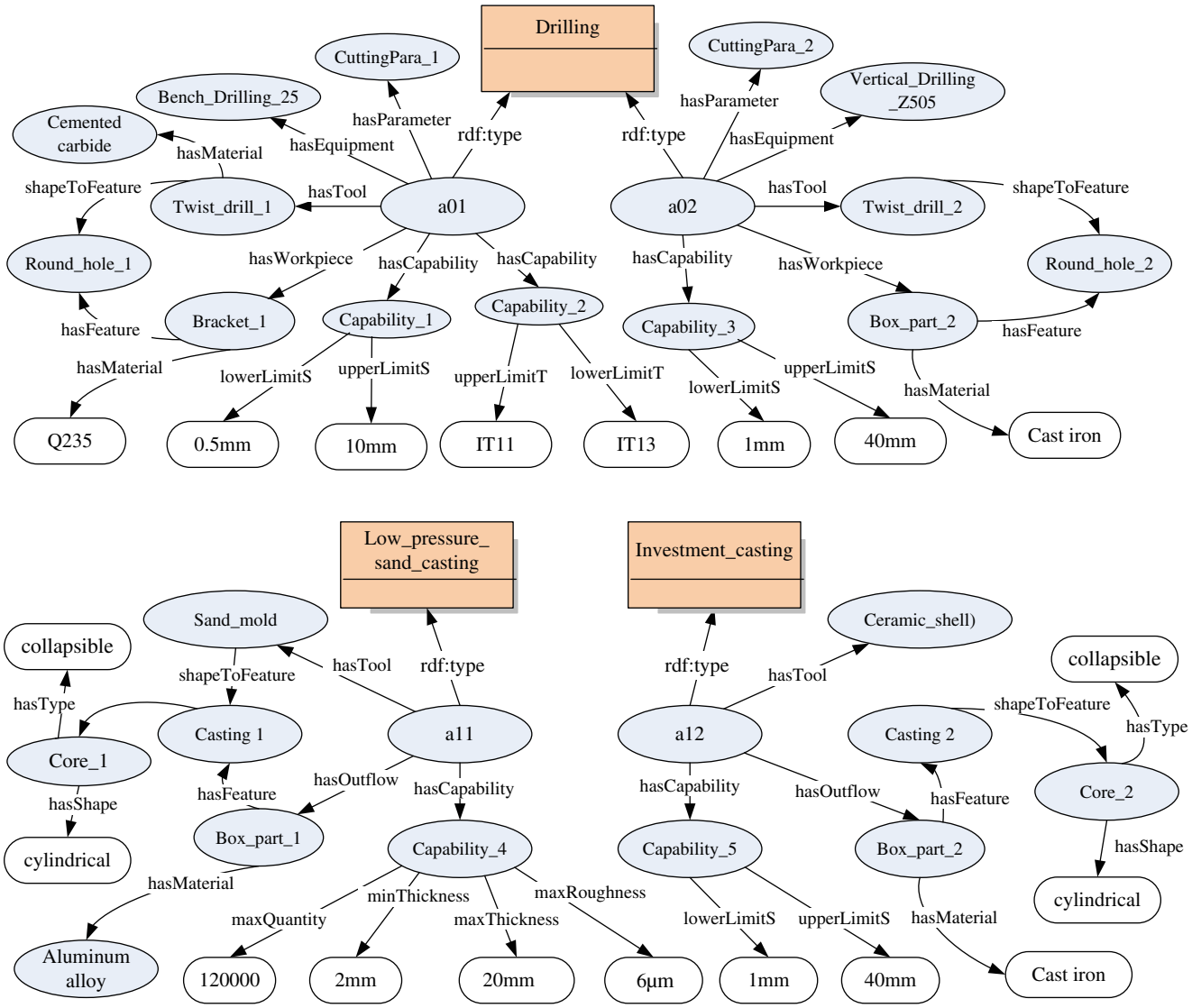


Fig. 11 Illustration of partial instances of the *Drilling* class and *Casting* class

cutting tools, molds, etc.). Therefore, we can obtain the following concept definitions:

$$\text{Shaping_Process}(x) \equiv \exists y(\text{Unit_Process}(x) \wedge \text{hasTool}(x,y))$$
$$\text{Non_Shaping_Process}(x) \equiv \exists y(\text{Unit_Process}(x) \wedge \neg\,\text{hasTool}(x,y))$$

5.1.2 Mass-conserving processes

First, the mass-conserving process is a shaping process. Secondly, the mass of an output product equals to the mass of an input material after the workpiece passes a unit process. A mass-conserving process concept, *Mass_Conserving*, can be defined as follow:

$$\begin{aligned}\text{Mass_Conserving}(x) \equiv \exists y,z(&\text{Shaping_Process}(x) \wedge \text{hasInflow}(x,y) \\ &\wedge \text{Material}(y) \wedge \text{hasOutflow}(x,z) \wedge \text{Product}(z) \\ &\wedge \exists m1,m2(\text{hasMass}(y,m1) \wedge \text{hasMass}(z,m2) \wedge (m1 = m2)))\end{aligned}$$

where x is a mass-conserving process, and y is a substance flow that inputs into the unit process x, and z is a substance flow that outputs from the unit process x. In addition, $m1$ and $m2$ are the mass of y and z, respectively.

5.1.3 Mass-reducing processes

In mass-reducing processes, a part of the input material's mass is lost and converted to waste (e.g., scrap) or other co-products. The mass of the output product is less than the mass of the input material after the workpiece passes a unit process. A mass-reducing process concept, *Mass_reducing*, can be defined as follows:

$$\begin{aligned}\text{Mass}-\text{Reducing}(x) \equiv \exists y,z(&\text{Shaping_Process}(x)\wedge\text{hasInflow}(x,y)\wedge\text{Material(y)}\\ &\wedge\ \text{hasOutflow}(x,z)\wedge\text{Product}(z)\wedge\exists m1,m2(\text{hasMass}(y,m1)\\ &\wedge\ \text{hasMass}(z,m2)\wedge(m2<m1)))\end{aligned}$$

5.1.4 Joining processes

Joining process is a unit process that makes two individual components be joined together. A joining process concept can be defined as follows:

$$\begin{aligned}\text{Joining}(x)\equiv\exists y1,y2,z(&\text{Shaping_Process}(x)\wedge\text{hasInflow}(x,y1)\wedge\text{Material}(y1)\\ &\wedge\text{hasInflow}(x,y2)\wedge\text{Material}(y2)\wedge\text{hasOutflow}(x,z)\wedge\text{Product}(z)\\ &\wedge\exists m1,m2,\text{m3}(\text{hasMass}(y1,m1)\wedge\text{hasMass}(y2,m2)\wedge\text{hasMass}(z,m3)\\ &\wedge(m3=m1+\text{m2})))\end{aligned}$$

In the product assembly practice, the joining processes are usually implemented by specific joining methods. Currently, the typical mechanical joining methods include mechanical fastening, welding, soldering, gluing, and so on. This paper defines the joining process based on these methods as the subclasses of the *Joining* class, namely the *Mechanical_fastening* class, *Welding* class, *Soldering* class, and *Gluing* class.

The important characteristic of joining processes is that the assembling joints need a new third material (e.g., bolt, weld, etc.) to occupy a part of space of joining components. Currently, using Smith's mereotoplogy [26] to define joining processes formally should be a feasible method.

5.1.5 Cutting processes

The cutting process is a major machining process to create shapes on workpiece surfaces. There are many cutting processes available in industries. The important characteristic of a cutting process is that it is a mass-reducing process that uses a specific cutter to shape a workpiece. The following gives the definitions of two typical cutting processes, turning process, and milling process.

$$\begin{aligned}\text{Cutting}(x) &\equiv \text{Mass_Reducing}(x)\wedge\exists\text{y}(\text{hasTool}(x,y)\wedge\text{Cutting_Tool}(y))\\ \text{Turning}(x) &\equiv \text{Cutting}(x)\wedge\exists\text{y}(\text{hasTool}(x,y)\wedge\text{Turning_Tool}(y))\\ \text{Milling}(x) &\equiv \text{Cutting}(x)\wedge\exists\text{y}(\text{hasTool}(x,y)\wedge\text{Milling_Tool}(y))\end{aligned}$$

5.2 Shaping rules for unit processes

The shape is a geometrical description of an object abstracting from the geometrical boundary elements (e.g., point, line, surface). The shaping process converts the shape of an input workpiece into the shape of an output product. The shaping knowledge rules define the causal relationships between the shape changes and unit process, equipment, tools, etc.

1. Shaping rules for the drilling process.

In general, the drilling process mainly produces a hole feature on solid material surfaces. The hole feature can be classified into the round hole, threaded hole, center hole, etc. According to using different cutting tools, shaping rules can be defined. The following gives the definitions of shaping rules for the drilling process:

- Round hole feature.

$$\begin{aligned}\forall x,f1,f2:(&\text{Material_Solid}(f1)\wedge\text{Round_Hole}(f2)\leftarrow\exists t(\text{Drilling}(x)\wedge\text{hasTool}(x,t)\\ &\wedge\text{Twist_drill}(t)\wedge\text{shapeFromFeature}(t,f1)\wedge\text{shapeToFeature}(t,f2)))\end{aligned}$$

where x, t, $f1$, $f2$ denotes a drilling process, a twist drill, a solid feature, and a round hole feature, respectively.

- Threaded hole feature.

The manufacture of a threaded hole feature usually needs two steps. The first step is to drill a through hole or blind hole, then produce a threaded hole feature.

$$\forall x, f1, f2 : (\text{Round_Hole}(f1) \wedge \text{Thread_Hole}(f2) \leftarrow \exists t(\text{Drilling}(x) \wedge \text{hasTool}(x, t) \wedge \text{Tap_drill}(t) \wedge \text{shapeFromFeature}(t, f1) \wedge \text{shapeToFeature}(t, f2)))$$

where $x, t, f1, f2$ denotes a drilling process, a tap drill, a round hole feature, a threaded hole feature, respectively.

2. Shaping rules for the boring process.

In general, the boring process mainly aims to enlarge the size or get much higher accuracy for an existing hole. The shaping rules can be defined as follow:

$$\forall x, f1, f2 : (\text{Round_Hole}(f1) \wedge \text{Round_Hole}(f2) \leftarrow \exists t(\text{Boring}(x) \wedge \text{hasTool}(x, t) \wedge \text{Boring_cutter}(t) \wedge \text{shapeFromFeature}(t, f1) \wedge \text{shapeToFeature}(t, f2)))$$

3. Shaping rules for the turning process.

The turning process can make more shape features on workpiece surfaces than the drilling process. The main shape features produced by the turning process are the cylindrical surface, the tapered surface, the threaded surface, the ring groove, etc. The following gives the definitions of shaping rules for the cylindrical surface and the round hole:

$$\forall x, f1, f2 : (\text{Cylindrical_surface}(f1) \wedge \text{Cylindrical_surface}(f2) \leftarrow \exists t(\text{Turning}(x) \wedge \text{hasTool}(x, t) \wedge \text{External_turning_tool}(t)) \wedge \text{shapeFromFeature}(t, f1) \wedge \text{shapeToFeature}(t, f2)))$$

$$\forall x, f1, f2, f3 : (\text{Material_Solid}(f1) \wedge \text{Round_Hole}(f2) \wedge \text{Round_Hole}(f3) \leftarrow \exists t1, t2(\text{Turning}(x) \wedge \text{hasTool}(x, t1) \wedge \text{Twist_drill}(t1)) \wedge \text{shapeFromFeature}(t1, f1) \wedge \text{shapeToFeature}(t1, f2) \wedge \text{hasTool}(x, t2) \wedge \text{Inner_turning_tool}(t2) \wedge \text{shapeFromFeature}(t2, f3) \wedge \text{shapeToFeature}(t2, f3)))$$

5.3 Process scheme rules for unit processes

In the manufacturing process, it is difficult for a single unit process to complete the production of a feature with high quality requirements, so it is required to combine multiple unit processes sequentially to form a manufacturing route, or in the same unit process to change the cutting parameters to obtain the best machining quality and efficiency. This combination of an ordered set of unit processes is called the process scheme. Each process scheme employs different process configurations and has different manufacturing capabilities, so it is the important manufacturing process knowledge and widely used in the process planning.

As mentioned above, this paper employs SWRL as the knowledge base rule language. A SWRL rule contains an antecedent part, which is referred to as the body, and a consequent part, which is referred to as the head. We can define the process scheme rules by using SWRL, two types of which are defined as follows:

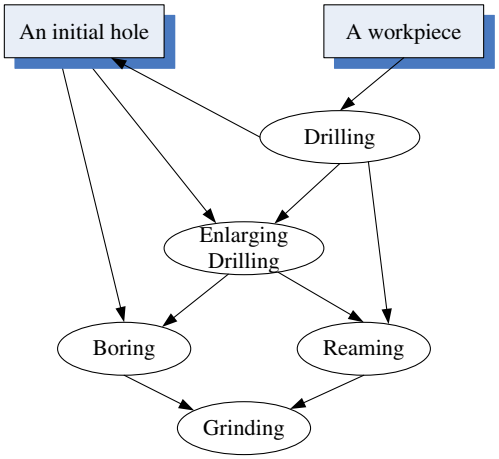


Fig. 12 The manufacturing routes of a hole

1. Process scheme rules based on manufacturing routes.

A feature can be machined by a number of feasible manufacturing routes. Each route is an ordered set of unit processes capable of producing the feature form (geometry and topology) and satisfying its technological requirements. For example, in order to produce a hole feature, there exist many feasible manufacturing routes as shown in Fig. 12. Some process schemes formed by the feasible and effective manufacturing routes should be defined in the knowledge base in order to be selected in the coming process planning.

According to the expert's knowledge, some feasible and effective process schemes can be defined as a set of new unit process concepts by defining them as the subclasses of the *Unit_Process* class. To illustrate the method, the following gives two definitions of process schemes for manufacturing a round hole feature.

- Drilling–enlarging process scheme.

In the drilling–enlarging process scheme, a twist drill and an enlarging hole drill are employed, which is defined as follow.

$$\text{Unit_Process}(?x) \wedge \text{hasTool}(?x, ?t1) \wedge \text{Twist_drill}(?t1) \wedge \text{hasTool}(?x, ?t2) \wedge \text{Enlarging_drill}(?t2) \rightarrow \text{Drilling_Enlarging}(?x)$$

- Drilling–enlarging–reaming process scheme.

In the drilling–enlarging–reaming process scheme, a twist drill, an enlarging hole drill, and a reaming tool are employed, which is defined as follow.

$$\text{Unit_Process}(?x) \wedge \text{hasTool}(?x, ?t1) \wedge \text{Twist_drill}(?t1) \wedge \text{hasTool}(?x, ?t2) \wedge \text{Enlarging_drill}(?t2) \wedge \text{hasTool}(?x, ?t3) \wedge \text{Reaming_tool}(?t3) \rightarrow \text{Drilling_Enlarging_Reaming}(?x)$$

After the process schemes have been defined, the instances of new defined unit processes need to be constructed so as to

provide knowledge queries. The method to construct instances can be referred to Section 4.4.

2. Knowledge rules based on process parameters.

In the same unit process, due to using different process parameters, their manufacturing quality and efficiency are not same. For example, a turning process is often classified into rough turning, semi-fine turning, and fine turning. The rough turning can obtain the higher machining efficiency and lower machining quality, so it is suitable for initial processing. In general, the classification depends on the process parameters. In the machining process, the process parameters mainly include the cutting speed, feed rate, and depth of cut. The process schemes formed by using different process parameters also can be defined as new concepts. To illustrate the definition method, a concept of rough turning is defined as follow.

In fact, it is difficult to determine the cutting parameters of a rough turning as it also depends on many other factors, such as the workpiece material, tools. The following example only illustrates the method, in which the material of the turning tool is cemented carbide, and the cutting speed is less than 160 m/min, and the cutting feed is greater than 0.2 mm/rev, and the depth of cut is greater than 2 mm.

$$\begin{aligned}&\mathrm{Turning}(?x)\wedge\mathrm{hasTool}(?x,?t)\wedge\mathrm{Cutting_Tool}(?t)\wedge\mathrm{Tool_Material}(?t,?m)\\&\wedge\mathrm{swrlb:stringEqualIgnoreCase}(?m,\mathrm{Cemented\ carbide})\\&\wedge\mathrm{hasParameters}(?x,?p)\wedge\mathrm{Cutting_Parameters}(?p)\\&\wedge\mathrm{Cutting_Speed}(?p,?v)\wedge\mathrm{swrlb:lessThan}(?v,160)\wedge\mathrm{Cutting_Feed}(?p,?f)\\&\wedge\mathrm{swrlb:greaterThan}(?f,0.2)\wedge\mathrm{Cutting_Depth}(?p,?a)\\&\wedge\mathrm{swrlb:greaterThan}(?a,2)\rightarrow\mathrm{Rough_Turning}(?x)\end{aligned}$$

6 Knowledge retrieval for unit processes

In a knowledge system, the capability of retrieving and reasoning relevant knowledge is the fundamental requirements. In practice, the so-called reasoning is that when a query is given, the knowledge-based system should be able to find and extract required implicit information from the knowledge base (ontologies, instances, and rules) built-in the system. Currently, SQWRL, a SWRL-based query language, has been widely used. SQWRL takes a standard SWRL rule antecedent and effectively treats it as a pattern specification for a query. An implementation of SQWRL has been developed in the SWRLTab plugin [25] in Protégé-OWL [27]. The implementation provides a JDBC-like Java interface to execute SQWRL queries in Java applications.

Based on the above analysis, in the reasoning layer of the knowledge representation framework, SQWRL is employed as the query language, and the Protégé-OWL ontology development toolkit supporting Jess [28] is adopted as a software implementation tool, which provides good supports for SWRL/SQWRL.

There are mainly three types of the knowledge retrieval for unit processes: (1) retrieving a unit process from multiple conditions, (2) querying the manufacturing capabilities of a given unit process, and (3) inferring a suitable unit process or process scheme for a given machining feature.

6.1 Querying unit processes

Retrieving a unit process from the knowledge base is one of the knowledge retrieval tasks. There are mainly the following retrieving approaches:

1. Querying all unit process classes in the knowledge base.

The taxonomy of unit process ontologies has been defined in the TBox of the knowledge base. Using the TBox query library with SQWRL query operators allows the writing of queries that ask questions about the classes and properties in an OWL ontology. Therefore, the retrieval of all unit processes can be implemented by the following query:

$$\mathrm{tbox:subClassOf}(?subClass,\mathrm{Unit_Process})\rightarrow\mathrm{sqwrl:select}(?subClass)$$

where Unit_Process denotes the base class to be retrieved. This query will return a retrieved result table in which all direct subclasses of the *Unit_Process* class are listed.

2. Querying unit processes from the process elements.

As already mentioned in Section 3, the main basic elements in a unit process are the equipment, workpiece, and tool. It is often required to retrieve a unit process from the basic process elements. For example, if all unit processes for processing sheet parts are needed to be retrieved, the following query needs to be executed and returns a table in which all unit processes that can process sheet part are listed.

$$\mathrm{Unit_Process}(?p)\wedge\mathrm{hasWorkpiece}(?p,?w)\wedge\mathrm{Sheet_Part}(?w)\rightarrow\mathrm{sqwrl:select}(?p)$$

This query will return a retrieved result table in which the unit process instances for processing sheet part are listed.

3. Querying unit processes from the given manufacturing requirements

In the process planning, if some manufacturing requirements, such as the machining feature, the accuracy grade, and the roughness, are known, it is often required to retrieve or select a suitable unit process. The following gives two query examples.

- Querying unit processes for making a round hole on sheet part.

$$\text{Unit_Process}(?p) \wedge \text{hasWorkpiece}(?p, ?w) \wedge \text{hasTool}(?p, ?t) \wedge \text{shapeToFeature}(?t, ?f) \wedge \text{Round_Hole}(?f) \wedge \text{Sheet_Part}(?w) \wedge \text{hasFeature}(?w, ?f) \rightarrow \text{sqwrl} : \text{select}(?p)$$

- Querying unit processes for machining a cylindrical surface of a shaft part.

The tolerance grade requires the lower limit is less than or equal to IT9 and the upper limit is equal to or greater than IT7. The roughness requires that the upper limit is less than or equal to 5 μm, and the lower limit is equal to or greater than 2.5 μm. The following query needs to be executed.

$$\begin{aligned}&\text{Unit_Process}(?p) \wedge \text{hasTool}(?p, ?t) \wedge \text{shapeToFeature}(?t, ?f) \wedge \text{Cylinder}(?f) \\ &\wedge \text{hasCapability}(?p, ?c) \wedge \text{Tolerance_Range}(?c) \wedge \text{hasCapability}(?p, ?c1) \\ &\wedge \text{Roughness_Range}(?c1) \wedge \text{lowerLimitT}(?c, ?lower) \\ &\wedge \text{swrlb} : \text{lessThan}(?lower, 9) \wedge \text{upperLimitT}(?c, ?upper) \\ &\wedge \text{swrlb} : \text{greaterThan}(?upper, 7) \wedge \text{lowerLimitR}(?c1, ?lo) \\ &\wedge \text{swrlb} : \text{lessThan}(?lo, 5) \wedge \text{upperLimitR}(?c1, ?up) \\ &\wedge \text{swrlb} : \text{greaterThan}(?up, 2.5) \rightarrow \text{sqwrl} : \text{select}(?p)\end{aligned}$$

This query will list all unit processes which can satisfy above technical requirements from the knowledge base, including the defined process schemes.

6.2 Querying manufacturing capabilities of a unit process

Querying manufacturing capabilities of a unit process is an important knowledge activity and mainly includes the following two aspects:

1. Querying the shaping capabilities.

The shape feature that a unit process can produce is related by the *shapeToFeature* property of the *Tool* class, so the following query needs to be executed.

$$\text{Unit_Process}(?p) \wedge \text{tbox} : \text{equalTo}(?p, \text{Qp}) \wedge \text{hasTool}(?p, ?t) \wedge \text{shapeToFeature}(?t, ?f) \rightarrow \text{sqwrl} : \text{select}(?f)$$

where Qp is a given unit process to be queried. This query will list all machining features that the unit process Qp can produce.

2. Querying the machining accuracy.

$$\text{Unit_Process}(?p,) \wedge \text{tbox} : \text{equalTo}(?p, \text{Qp}) \wedge \text{hasCapability}(?p, ?c) \wedge \text{Tolerance_Rang}(?c) \wedge \text{upperLimit}(?\text{c}, \text{t1}) \wedge \text{lowerLimit}(?\text{c}, ?\text{t2}) \rightarrow \text{sqwrl} : \text{select}(?t1, ?t2)$$

This query will list the upper limit and lower limit of the tolerance range that the unit process Qp can achieve.

7 An example for casting process selections

In order to demonstrate the proposed knowledge representation methodology, we choose the casting process selections of an automotive gearbox shown in Fig. 13 as the demonstrated example. The material of the casting is aluminum alloy and other technical requirements are shown in Table 2 [29]. It is evident that selecting a suitable casting process requires extensive knowledge of various casting processes. Based on the knowledge representation framework presented in this paper, we can easily realize the process selection as follows.

7.1 Constructing the casting process knowledge base

As mentioned above, the ontology-based process knowledge base includes the TBox, ABox, and SWRL rules. According to the knowledge representation framework presented in this paper, we can easily construct the knowledge representation of the casting process as follows.

1. Defining the casting process concepts.

Based on the unit process ontology mentioned above, 20 casting processes which have been widely used in the industry are defined as the subclass of the *Casting* class ontology, which is shown in Fig. 14.

2. Knowledge construction for the casting processes.

In the ontology knowledge base, it is required to construct the knowledge instances of the defined casting process concepts. We can consult casting experts, reference manuals, books, and other information to collect and capture the casting process knowledge. Then, the captured knowledge is input to the knowledge system by human–computer interaction so as to construct instances.

According to the method introduced in Section 4.4, the instances of above 20 casting process concepts are constructed in the ABox. For each casting process concept, it is required to create at least one instance. For some process concepts, according to the captured knowledge, multiple instances may be created, for example, in the same casting process, due to different casting materials, equipment, and molds, the manufacturing capabilities are different, so multiple different instances should be created for the same casting process concept.

7.2 Knowledge retrieval for casting processes

Based on the knowledge base of casting processes, it is easy to retrieve the suitable casting process according to the technical requirements. The knowledge retrieval needs to implement the following two steps:



Fig. 13 An automotive gearbox casting

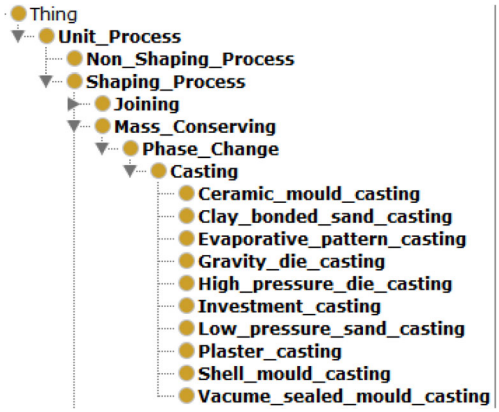


Fig. 14 Partial casting process classes

1. Creating a querying statement.

According to the defined ontology concepts and relationships, the technical requirements of this example can be represented as an SQWRL querying statement as shown in Fig. 15.

2. Executing the querying statement.

After the querying statement as shown in Fig. 15 is submitted to the SQWRL query engine, the querying statement is executed. The query engine will inference the unit process knowledge base and returns a table which lists the retrieved casting processes. From the instances of the retrieved casting processes, we can get information on casting process's name, mold, process parameters, etc. In this demonstrated example, the returned querying table lists four casting processes that can meet the technical requirements, which are same as the result of literature [29].

8 Implementation and discussion

8.1 Implementation of a prototype system

Based on the above-mentioned ontology-based knowledge representation model and framework, we have developed a software prototype system for the unit process knowledge representation. The development environments include three parts: (1) Protégé 4.3 editor (http://protege.stanford.edu) which is used to construct the unit process ontology, (2) Protégé-OWL ontology development toolkit [27] supporting Jess [28], and (3) Eclipse Java development tools (JDT) (www.eclipse.org) which are used to develop the retrieval and interface programs.

The knowledge prototype system is shown in Fig. 16, which mainly includes the following two modules:

- Knowledge constructor.

The knowledge constructor is a module of constructing unit process knowledge base, which is carried out in Protégé 4.3 editor. Based on the knowledge representation model presented in this paper, knowledge engineers define the unit process concepts and relationships by collecting and capturing the unit processes knowledge in related fields. The specific equipment, tools, manufacturing process parameters, and manufacturing capabilities are defined as the instances corresponding to the concept. In addition, according to the implicit semantic knowledge and process knowledge, a set of

Table 2 Partial technical requirements of the gearbox casting

Casting feature	Casting accuracy	Production quantity
Minimum section thickness, 2.5 mm	Tolerance required, ±1 mm	Production quantity required, >100,000
Maximum section thickness, 12 mm	Surface roughness required, 8 μm	
Maximum length, 300 mm	Linear dimension along parting plane, 300 mm	
Core shape: cylindrical		
Core type: collapsible		
Minimum core diameter, 10 mm		
Number of parting planes required, 1		
Weight, 4 kg		

Casting(?*p*)∧hasOutflow(?*p*,?*w*)∧Casting_Product(?*w*)∧hasMaterial(?*w*,?*m*)∧swrlb:stringEqualIgnoreCase(?*m*, aluminum)∧hasCore(?*w*,?*c*)∧Core(?*c*)∧hasShape(?*c*,?*s*)∧Cylinder(?*s*)∧hasCoreType(?*c*,?*ct*)∧Collapsible(?*ct*)∧minDiameter(?*c*,?*d*)∧swrlb:lessThan(?*d*,10)∧hasCapability(?*p*,?*a*1)∧Casting_Capbility(?*a*1)∧minThickness(?*a*1, ?*t*1)∧swrlb:lessThan(?*t*1,2.5)∧maxThickness(?*a*1, ?*t2)*∧swrlb:greaterThan*(?t2,*12)∧hasQuantity(?*a*1,?*q*)∧swrlb:greaterThan(?*q*,100000)∧maxRoughness(?*a*1,?*r*)∧swrlb:lessThan(?*r*,8)∧minAccuracy(?*a*1,?*d*1)∧swrlb:greaterThan(?*d*1,-1)∧maxAccuracy(?*a*1,?*d*2)∧swrlb:lessThan(?*d*2,1)∧hasCapabIlity(?*p*,?*a*2)∧Weight_Range(?*a*2)∧upperLimitW(?*a*2,?w1)∧swrlb:greaterThan(?*w*1,4)∧hasCapability(?*p*,?*a*3)∧Size_Range(?*a*3)∧upperLimitS(?*a*3, ?*su*)∧swrlb:greaterThan(?*su*, 400)→swrlb:select(?*p*)

Fig. 15 A querying statement for a casting process selection

SWRL rules is defined. All of the definitions (including concepts, instances, and rules) are saved in a "Unit_Process.owl" file.

- Knowledge searcher.

The knowledge searcher is an application program for retrieving unit process knowledge, which is developed based on Eclipse Java development tools and Protégé-OWL API. The knowledge searcher program consists of the following two parts:

1. Knowledge query interface.

The knowledge query interface performs two tasks: One is to provide a human interface for manufacturing engineers to input query requests. We make full use of the knowledge base's information. In other words, most items of query requests are determined by retrieving the class and property in the knowledge base. The other is to convert the query requests of users into a SQWRL statement according to the SQWRL syntactic requirements and submit to the SQWRL query engine. In addition, the returned query results need to be processed and display on the interface.

2. Implementation of SQWRL query.

In order to implement the SQWRL query, a SQWRL query engine must be created at first. To create a SQWRL query engine two program steps are needed. Firstly, the OWL ontology knowledge base is loaded, and an OWL model is created, which usually requires a uniform resource address for the knowledge base file (Unit_Process.owl). Then, a SQWRL query engine is created by executing the constructor function of the SQWRLQueryEngineFactory class.

Once the SQWRL query engine has been created, the querying statement created from the knowledge query interface can be submitted to the SQWRL query engine and is executed. If the query can generate a result, the query engine will return a table with the column and row. The returned table is processed, and the querying results are sent back to the knowledge query interface. Partial program codes of creating a SQWRL query engine and processing the query table are as shown in Fig. 17.

8.2 Discussion

In the face of a large number of different types of manufacturing processes, to create a unified, open and shared and scalable process knowledge representation is a great

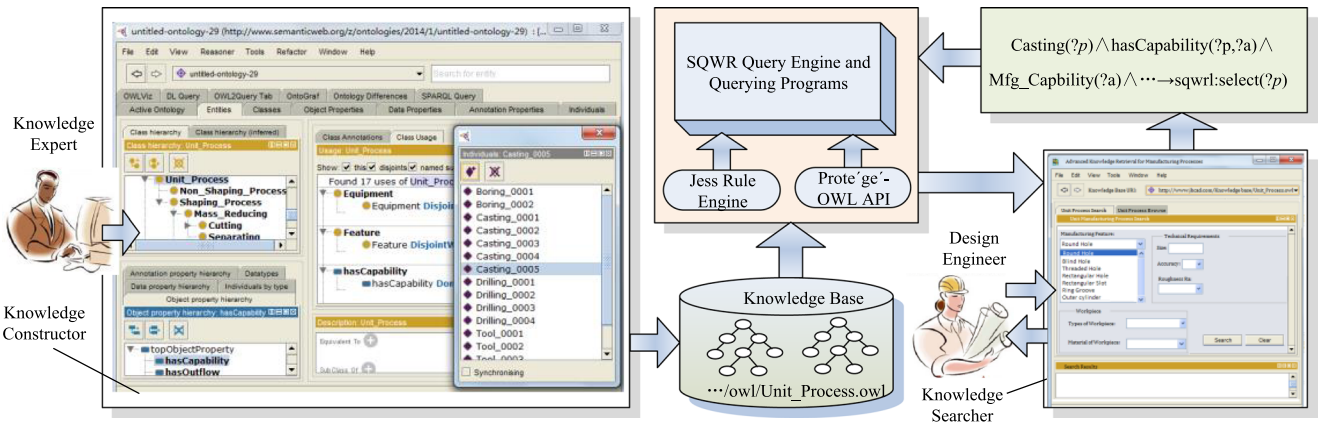


Fig. 16 Illustration of the knowledge prototype system for unit processes

```
……
OWLModel owlModel = ProtegeOWL.createJenaOWLModelFromURI(uri);  //Creating an OWL model; uri is the resource address for owl file
SQWRLQueryEngine queryEngine = SQWRLQueryEngineFactory.create(owlModel); //Creating a query engine
SQWRLResult result = queryEngine.runSQWRLQuery("Query-1", queryStatement); //queryStatement is a string Variable which is a query statement converted by knowledge querying for unit process
while (result.hasNext()) {
   DataValue Value1 = result.getDataValue("?name");  //Getting the name of a retrieved unit process
   DataValue Value2 = result.getDataValue("?tool"); //Getting the tool of a retrieved unit process
   ……
   result.next();
}
```

Fig. 17 Partial program codes for creating a SQWRL query engine

challenge. The ontology technology has much more advantages than other traditional modeling methodologies in the concept classification, sharing, and formalization. The works presented in this paper improve the knowledge representation of unit manufacturing processes from the following three aspects:

1. Establishing an ontology-based knowledge representation framework for unit manufacturing processes.

Firstly, we abstract and define nine top level classes and corresponding relationships, as well as knowledge rules encoded with SWRL, which provides the representation for a variety of manufacturing process knowledge at the conceptual level. Secondly, the specific knowledge of various unit processes is represented as a set of instances labeled by OWL classes. Moreover, the concept class, rules and instances can be combined well to implement knowledge inferences.

2. Improving the interoperability of the knowledge representation.

Firstly, we explicitly define the terminology and taxonomy of unit processes. Secondly, a formalized approach to define the taxonomic semantics of unit processes is presented. The taxonomic semantics of the unit process concepts is a kind of important implicit knowledge, but few studies have involved. This paper analyzes the characteristics of material flows, based on describe logic a formalized approach to define the taxonomic semantics of the unit processes is proposed. This work can help agents to automatically classify unit process instances and check the concept consistency so as to improve the interoperability of the knowledge representation

3. Improving the scalability of the knowledge representation system.

As a wide variety of unit manufacturing processes involves different domain knowledge, it is very important for a knowledge system to have a good scalability. This work improves the scalability of the knowledge representation from the following aspects:

1. The representation model provides a concept framework based on ontology.

Based on the concept framework, user can easily define their subclasses and properties on the base of base classes according to their domain knowledge requirements. For example, as introduced in Section 7, in order to represent the knowledge of casting processes we can load the knowledge base file to the Protégé editor, and define the new substances of the casting process (*Casting* class) and construct the knowledge instances of the new defined casting process concepts.

2. New unit process concepts and relations can be defined by SWRL rules.

Many new concepts resulting from causal relations (a kind of knowledge) can be defined by SWRL rules, which extends the knowledge representation capability greatly. For example, as mentioned in Section 5, the new concepts, such as rough turning, fine turning, and various process schemes, can be easily defined and combined into the original concept framework.

3. No need to modify the application program of the knowledge system.

Compared with traditional knowledge representation methods, modifying or extending the process knowledge is carried out in the Protégé editor and does not need to modify any application program.

In addition, the presented knowledge representation bases on the description logic. Description logic is a mathematically rigorous representation that enables reasoning to be performed on the ontology including concept consistency, concept equivalence, and concept subsumption.

The proposed knowledge representation for manufacturing unit process is only a part of manufacturing knowledge base system, which needs to be further extended and integrated into the CAPP system or the manufacturing service system so as to provide the intelligent decision support for process planning or process selections.

9 Conclusion

A unit manufacturing process is the foundation for the process planning and manufacturing services. In the face of a large number of commercially used unit processes and new emerging unit processes, it is very necessary to research and develop the knowledge representation and information processing approaches for unit processes. In this paper, the basic unit process concepts and relationships among concepts are analyzed in detail. A new ontology-based knowledge representation model for unit processes is proposed. The following are the contributed works:

1. The semantic Web technology is introduced into the knowledge modeling for unit processes, and a new ontology-based knowledge representation model and framework for unit processes are presented.
2. The inherent basic concepts in a unit process and the relationships between concepts are researched. Nine top level concept classes of unit processes are defined with OWL, which constitute the base classes of the knowledge representation for unit processes.
3. A formalization for the taxonomic semantics and the description of manufacturing capability knowledge is presented. A rule base that consists of a set of rules coded with SWRL is built, which is used to enable reasoning on the taxonomic semantics and manufacturing knowledge of unit processes.
4. A knowledge retrieval and reasoning approach that integrates OWL ontologies and SWRL rules for unit processes is proposed. Some knowledge retrieval examples employing SQWRL queries are given. A knowledge prototype system for unit processes is developed, which is a new attempt to make the process knowledge representation into applications.

The completed works in this paper lay a foundation for constructing an open and shared manufacturing process knowledge system. Further research works include as follows:

1. Based on the explicit and shared characteristics of ontology modeling, the research on Web-based and open knowledge modeling for unit process will be carried out, which is able to cope with the challenges that result from constantly emerging unit processes and heavily relying on the domain knowledge.
2. Based on the knowledge representation proposed in this paper, the research on the intelligent process planning and decision making based on knowledge will be carried out.

Acknowledgements This work is supported by the National Science Foundation of China (grant nos. 60773214 and 51375069). The authors thank anonymous reviewers for their helpful suggestions in this study.

References

1. National Research Council (1995) Unit manufacturing processes: issues and opportunities in research. National Academy Press, Washington D C
2. Xun X, Wang L, Newman ST (2011) Computer-aided process planning—a critical review of recent developments and future trends. Int J Comput Integr Manuf 24(1):1–31
3. Gay JC (2007) Introduction and applications of DIN 8580: following the German Standard DIN 8580 Fertigungsverfahren. Univerisitat di Girona Publications, Girona
4. Sormmaz DN, Khoshnevis B (1997) Process planning knowledge representation using an object-oriented data model. Int J Comput Integr Manuf 10(1–4):92–104
5. Hayes C (1989) Automating process planning: using feature interaction to guide search. J Manuf Syst 8(1):1–15
6. Chang T-C (1990) Expert process planning for manufacturing. Addison-Wesley, Menlo Park
7. Liu Z, Wang L (2007) Sequencing of interacting prismatic machining features for process planning. Comput Ind 58(4):295–303
8. Guerra-Zubiaga DA, Young RIM (2008) Design of a manufacturing knowledge model. Int J Comput Integr Manuf 21(5):526–539
9. Chen WL, Xie SQ, Zeng FF, Li BM (2011) A new process knowledge representation approach using parameter flow chart. Comput Ind 62(1):9–22
10. Xu H-M, Yuan M-H, Li D-B (2009) A novel process planning schema based on process knowledge customization. Int J Adv Manuf Technol 44(1–2):161–172
11. Hao Y-T, Ma J-Y (2006) A knowledge-based auto-reasoning methodology in hole-machining process planning. Comput Ind 57(4): 297–304
12. Todd RH, Allen DK, Alting L (1994) Fundamental principles of manufacturing processes. Industrial Press Inc., New York
13. Naish JC (1996) Process capability modeling in an integrated concurrent engineering system: the feature-oriented capability module. J Mater Process Technol 61(1–2):124–129
14. Gruber TR (1993) A translation approach to portable ontology specifications. J Knowl Acquis 5(2):199–220
15. Zhu LJ, Jayaram U, Kim O (2012) Semantic applications enabling reasoning in product assembly ontologies—moving past modeling. ASME J Comput Inf Sci Eng 12(1):011009
16. Gruninger M, Menzel C (2003) The Process Specification Language (PSL) theory and applications. AI Mag 24:63–74
17. Qiao LH, Kao ST, Zhang YZ (2011) Manufacturing process modelling using process specification language. Int J Adv Manuf Technol 55(5–8):549–563

18. McGuinness DL, Harmelen FV (2004) OWL Web ontology language overview. http://www.w3.org/TR/owl-features/ February 20
19. Mikosa WL, Ferreirab JCE, Boturac PEA, Freitas LS (2011) A system for distributed sharing and reuse of design and manufacturing knowledge in the PFMEA domain using a description logics-based ontology. J Manuf Syst 30(3):133–143
20. Garcia-Crespo A, Ruiz-Mezcua B, Lopez-Cuadrado JL, Gomez-Berbis JM (2010) Conceptual model for semantic representation of industrial manufacturing processes. Comput Ind 61(7):595–612
21. Lemaignan S, Siadat A, Dantan J-Y, Semenenko A (2006) MASON: a proposal for an ontology of manufacturing domain. In: Proceedings of the international IEEE workshop on distributed intelligent systems (DIS 2006), Prague, June
22. Martínez-Pellitero S, Barreiro J, Cuesta E, Álvarez BJ (2011) A new process-based ontology for KBE system implementation: application to inspection process planning. Int J Adv Manuf Technol 57(1–4): 325–339
23. Brachman RJ, Levesque HJ (2004) Knowledge representation and reasoning. Morgan Kaufmann, San Francisco
24. Horrocks I, Patel-schneider PF, Boley H, Tabet S, Grosof B, Dean M (2004) SWRL: a semantic Web rule language combining OWL and RuleML. http://www.w3.org/Submission/SWRL, July 20
25. O'Connor M, Das A (2009) SQWRL: a query language for OWL. In: Proceedings of the 5th International Workshop on OWL: Experiences and Directions (OWLED 2009), Chantilly, VA
26. Smith B (1996) Mereotopology: a theory of parts and boundaries. Data Knowl Eng 20(3):287–303
27. Knublauch H, Fergerson RW, Noy NF, Musen MA (2004) The Protégé OWL plugin: an open development environment for semantic Web applications. Lect Notes Comput Sci 3298:229–243
28. Friedman-Hill E (2013) Jess, the rule engine for the Java platform. http://www.jessrules.com/jess/index.shtml
29. Er A, Dias R (2000) A rule-based expert system approach to process selection for cast components. Knowl-Based Syst 13(4):225–234