



An Ontology for Industrial Intelligent Model Library and Its Distributed Computing Application

Cunnian Gao^{1,2}, Hao Ren^{2(✉)}, Wei Cui^{1,2}, Xiaojun Liang², Chunhua Yang^{2,3}, Weihua Gui^{2,3}, Bei Sun³, and KeKe Huang³

¹ School of Future Technology, South China University of Technology, Guangzhou, Guangdong, China

² Department of Strategic and Advanced Interdisciplinary Research, Peng Cheng Laboratory, Shenzhen, Guangdong, China

renh@pcl.ac.cn

³ School of Automation, Central South University, Changsha, Hunan, China

Abstract. In the context of Industry 4.0, the paradigm of manufacturing has shifted from autonomous to intelligent by integrating advanced communication technologies. However, to enable manufacturers to respond quickly and accurately to the complex environment of manufacturing, knowledge of manufacturing required suitable representation. Ontology is a proper solution for knowledge representation, which is used to describe concepts and attributes in a specified domain. This paper proposes an ontology-based industrial model and significantly improves the interoperability of the models. Firstly, we conceptualize the attribute of the industrial models by providing concept and their properties in the schema layer of the ontology. Then, according to the data collected from the manufacturing system, several instances are created and stored in the data layer. In addition, we present a prototype distributed computing application. The result suggests that the ontology can optimize the management of industrial models and achieve interoperability between models.

Keywords: Ontology · Knowledge representation · Industry 4.0 · Industrial model

1 Introduction

The integration of advanced communication technologies, e.g., cyber-physical systems (CPS), the Internet of Things (IoT), edge computing, and artificial intelligence, is guiding traditional manufacturing to next-generation intelligent manufacturing which is generally known as Industrial 4.0. The intelligent manufacturing system enables collecting data from sensors and making smart decisions based on real-time communication and collaboration with machines, thus, can improve the performance of the manufacturing system significantly. Meanwhile,

the large amount of data collected from manufacturing systems and knowledge of manufacturing processes are the cornerstone of intelligent manufacturing.

The most crucial aspect of intelligent manufacturing is the self-learning and self-decision-making capabilities enabled by manufacturing knowledge. In general, knowledge plays a crucial role in providing necessary information and controlling the manufacturing system automatically. On the one hand, knowledge of data analysis generated by machine learning and data visualization tool can assist manufacturing systems in identifying the status and trends of manufacturing processes. On the other hand, based on the knowledge of artificial intelligent, manufacturing systems can predict equipment failures and fix them timely, thus, production lines can avoid unnecessary work and reduce production costs. In this context, the system can ensure that the products meet the required quality standards and improve production efficiency. Nowadays, knowledge modeling has attracted much attention from academia and industry. H.L. Wei et al. [1] focused on mechanistic models of additive manufacturing for predicting experimental observations correctly. To provide an advanced solution for the smart factory, M. Ghahramani et al. [2] deployed machine learning and data analysis technologies in semiconductor manufacturing processes. However, due to different development standards among engineers, e.g., the meaning of symbols and programming language, most models are not interoperable. Moreover, these models are scattered across different domains, thus, it is a time-consuming and high-cost task that searching for the specific model from different model libraries and deploying it in the manufacturing system. In addition, intelligent manufacturing is a complex domain that is comprised of different concepts, and it is a challenging task that represents manufacturing knowledge in a generic way. Hence, there is an urgent demand for designing an appropriate knowledge representation mechanism for intelligent manufacturing.

Ontology is a proper solution to achieve interoperability and knowledge representation, which maintains the machine-interoperable concepts in a specific domain and the relation among them by providing a standard platform that supports data process and information exchange. [3] Since ontology represents concepts and relationships in a semantic way, the ontology-based model can be applied in various domains and enables semantic interoperability. Nowadays, various ontology-based models have been developed to represent and share manufacturing knowledge. Järvenpää et al. [4] developed an OWL-based manufacturing resource capability ontology (MaRCO), which describes the functionality and constraint of manufacturing resource and assists the system in adjusting to the change in manufacturing system. Dinar et al. [5] developed an additive manufacturing ontology, which stores the concept of domain and experiential knowledge of additive manufacturing and provides guidelines for designers. A new manufacturing ontology was developed by Saha et al. [6] to model manufacturing operations and sequencing knowledge and the result of the experiment suggests that the model can greatly facilitate manufacturing engineers in making decisions. In order to achieve the goal of reasoning automatically and assisting decision-making, Sanfilippo [7] and his colleagues reused the Descriptive Ontology for

Linguistic and Cognitive Engineering (DOLCE) ontology and extended several branches by adding concepts related to the additive manufacturing process. However, most previous works focus on conceptualizing manufacturing processes, equipment, and operations, while ignoring the knowledge of reaction mechanisms during the manufacturing processes. Moreover, most models developed in previous work are only applicable to specific domains. To the best of our knowledge, there is no production mechanism model involved in manufacturing processes.

This paper is aimed at developing an ontology-based industrial intelligent manufacturing model to fill this gap. In this paper, a new industrial intelligent manufacturing ontology is developed to model knowledge of manufacturing processes and make it easily sharable. Due to the explicit concept in the proposed ontology, the model is suitable for various applications. Firstly, our ontology is comprised of two layers: the schema layer and the data layer. We conceptualize basic information of the model and provide the properties of every concept in the schema layer. Then, according to the data of the industrial model, a number of instances are created and stored in the data layer. Furthermore, to verify the applicability and interoperability of the proposed ontology, we experiment by applying the ontology to distributed computing. The result suggests that our proposal is feasible and the ontology-based model is easy to manage. The knowledge of manufacturing processes is represented semantically, thus, users without expertise in a specific domain can quickly obtain useful information about the model. The contribution of this paper is listed as follows:

- Develop a new ontology-based intelligent manufacturing model by extending the model into two layers, the former layer is the schema layer which is comprised of the concepts of the industrial model and relationships between different entities, and the latter layer is the data layer, which contains the instances of the ontology and stored data collected from manufacturing system.
- To the best of our knowledge, we are the first to develop an ontology-based industrial model library for wet-process zinc smelting. The number of models in our library is more than 100 and will continue to grow in the future. Moreover, we map the ontology into the neo4j database and visualize the relationship between models explicitly, a further benefit is that the model can be classified by machine learning algorithms.
- Conduct a prototype experiment in distributed computing application and validate the interoperability characteristic of the proposed ontology. Due to the explicit concept of the model, the edge node can quickly obtain the main information of the model, e.g., the functionality of the model, and the usage of the model, thus, the time consumption of distributed computing is decreased significantly.

The rest of the paper is organized as follows: Sect. 2 reviews the state-of-the-art ontology development and summarizes the shortcomings of previous works and the challenges we are facing. Section 3 not only demonstrates the development of a new ontology-based industrial intelligent model in detail but also presents the construction and visualization of the industrial intelligent model

library. Section 4 presents the application of our proposal and the experiment suggests that the proposed ontology is feasible and interoperable. The conclusion of this paper and a roadmap for research directions for future works are outlined in Sect. 5.

2 Related Work

The Semantic Web, coined by Tim Berners-Lee, is an extension of the World Wide Web, which aims to encode the data and make it machine-readable [13]. To encode the semantic data and represent knowledge from heterogeneous resources, the WWW Consortium (W3C) develops the Resource Description Framework and Web Ontology Language(OWL) technologies. Indeed, ontology plays an essential role in representing knowledge on the Semantic Web and making it understandable to electronic agents. In this section, we review the existing ontology developed for intelligent manufacturing. Then we conclude the challenge we are facing.

2.1 Ontology for Intelligent Manufacturing

Ontology is an intelligent solution to capture knowledge in a specific domain and enable machines to reason the change in complex environments. In particular, integrating ontology in manufacturing can improve the intelligence level of the manufacturing system, and a great number of previous works have developed ontologies for different purposes. For instance, Lemaignan et al. [8] developed a manufacturing semantics ontology(MASON) for capturing the semantics of manufacturing operations. López et al. [11] created a chemical ontology to represent chemical substances and elements. Farazi et al. [12] encapsulated the chemical kinetic reaction mechanism into the ontology and created instances to construct a knowledge base. To validate the accuracy and correctness of the ontology, they applied the ontology for chemical kinetic reaction mechanisms (OntoKin) to three use cases, which include querying across mechanisms, modeling atmospheric pollution dispersion, and a mechanism browser tool. Although the result indicates that chemical ontology can help chemical knowledge exchange, they focus on combustion chemistry particularly.

In another study, to achieve flexibility and adaptability in Cyber-Physical Production Systems (CPPS), Günter Bitsch et al. [14] created an Intelligent Manufacturing Knowledge Ontology Repository (IMKOR), which addresses the interoperability issue by connecting different domain ontologies. According to the manufacturing process planning document(STEP-NC), Zhao et al. [15] developed a self-learning ontology for representing the capability of the machine in cloud manufacturing, the core method is the combination of algorithm and analysis result. However, while the common purpose of previous works is to overcome the interoperability issue, most of them focus on containing concepts in a specific domain as much as possible and ignore the practicality of the ontologies. In this context, we develop an ontology-based approach for the standard representation

of the industrial models to enable interoperability and flexibility. Meanwhile, by maintaining the industrial model library, manufacturers can fetch the models that they are searching for.

2.2 Open Challenges

The review of previous works exposes three main challenges concerning intelligent manufacturing ontology. From the application perspective, most existing ontologies are designed for a specific domain, thus, it is difficult to reuse the existing ontology to model knowledge of another domain. In addition, most of the previous works focus on conceptualizing manufacturing operations, materials, and equipment, while lacking knowledge of reaction mechanisms. Finally, the existing ontologies are difficult to manage, and it is time-consumption work to find the relevant ontologies and reuse them to reduce development costs and avoid ontology redundancy. In this context, we aim to address these issues by developing a new ontology-based industrial intelligent model for manufacturing knowledge representation and model management.

3 The Development of the Ontology

Ontology is useful for knowledge representation and sharing by enabling interoperability. There are several methodologies for developing ontology, such as Integrated Definition for Ontology Description Capture Method(IDEF5) [10], CyC [9]. In this paper, we choose the IDEF5 methodology for developing the proposed ontology. The reason we choose this methodology is that it can capture the real-world object in an intuitive and natural form. In this section, we introduce ontology development, which includes two layers. Section 3.1 presents the schema layer of the ontology, and the data layer is introduced in Sect. 3.2.

3.1 The Schema Layer of the Ontology

In the schema layer, we conceptualize the attribute of the industrial models by provide concept and their properties. To guarantee the semantics of ontology, we choose Web Ontology Language(OWL) as the ontology encoding language, because OWL can recognize various formats of data, e.g., XML, RDF, JSON-LD, and turtle syntax. Meanwhile, we take Protégé 5.5.0 as the ontology editor and visualize the structure of the ontology by using the OntoGraf plug-in. Protégé 5.5.0 is a useful open-source ontology editor tool, which is developed by Stanford University. The key steps required in creating an ontology as follows.

1. identify the key concepts of the industrial model and design the hierarchical relationships between classes by using Protégé 5.5.0.
2. identify the attribute of the concepts, including data properties and object properties. The object property represents the relationship between concepts, while the data property connects different formats of attribute data with the concepts (e.g. the value of the input variable.).

To ensure the interoperability of the ontology, designing the schema layer is the first step in developing the ontology. The structure of the schema layer is shown in Fig. 1.

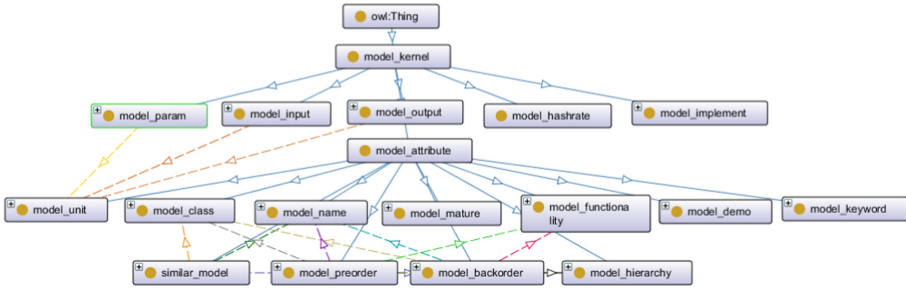


Fig. 1. The structure of the schema layer of the industrial models. The solid arrows represent the inheritance between the classes, and the dotted arrows represent the object properties.

The core concepts and description of the industrial model are given in Table 1 and Table 2 reports the properties used in the ontology. For instance, **model_kernel** is the main class of the model. Furthermore, **model_input**, **model_output**, and **model_param** are the subclasses of **model_kernel**. For instance, as for a manufacturing process status prediction model based on Long Short-Term Memory (LSTM), some features of the system represents the input of the model and the status prediction represents the output. Meanwhile, some hyperparameters (e.g. learning rate, epoch, batch size) are denoted by **model_param**. In order to enhance the machine-readability of the ontology, some basic attributes information of the industrial models is classified into different classes, which inherit their superclass **model_attribute**. The class **model_class** declares the class of industrial model, e.g., material type, industrial process, or industrial equipment. The maturity level of the model indicates the quality of the model, the machine can judge the maturity level according to the instance of the class **model_mature**. Similarly, the class **model_functionality** tells the machine the main functionality and usage of the model and the class **model_keyword** indicates which fields the model is related to. In the process of industrial manufacturing, many processes are interrelated, and the output of the previous process will be the input of the next process. Therefore, the class **model_preorder**, **model_backorder**, **similar_model** and **model_hierarchy** is designed to show the model id to get the input variable from the previous process or similar process.

The models can be implemented in a variety of ways. One of the important features is the inconsistency of the programming language (e.g. Python, Golang, and Java), which significantly hinders the interoperability of the models. The class **model_implement** is responsible for maintaining the interoperability of the models. The value of **data_property** *code_language* is the programming

Table 1. Description of main classes and their subclasses in industrial ontology

Class	Description
model_attribute	“A series of attributes including the basic information of the models.”
model_class	“According to the application scenario, divide the models into different classes.”
model_name	“The unique identity of the model.”
model_functionality	“The functionality of the model.”
model_keyword	“The labels of the model, which facilitate retrieval of models from model libraries.”
model_unit	“The dimensional unit of a variable.”
model_demo	“A test version of the model implement.”
model_mature	“The development level of the model.”
model_preorder	“The model that is inherited by this model.”
model_backorder	“The model that inherits this model.”
similar_model	“The model equipped similar functionality.”
model_hierarchy	“There is a cascading inheritance relationship between models, the hierarchical dimension of the model is beneficial for indexing the interconnected models.”
model_input	“Input variable of the model.”
model_output	“Result of computation.”
model_param	“The parameters of the model.”
model_implement	“Detail information of the implement, such as the coding, programming language, the contributors, etc.”
model_hashrate	“Detail information of the computation, such as the IP or the API of cloud computing.”

language, and the machine can pull a related docker to run the code, which can get from the property *code_description*.

Finally, accelerating the computational speed of the models is another purpose of ontology. Distribute computing is an excellent solution, which has received considerable attention in the past ten years. The **model_hashrate** is a general class for recording the running results of the model.

3.2 The Data Layer of the Ontology

In the data layer, according to different mechanisms, we create the individual instances of the classes defined in the schema layer. All of the data and basic information are stored in this layer. For instance, we can create an individual instance *Implement1* to represent the implement of the model, which belong to the class **model_implement**. This instance has the properties value as follows.

Table 2. The properties used in the ontology

Property	Domain	Range	Description
has_id	model_input, model_output, model_param, model_implement, model_hashrate, model_class	int	“The unique identity of the instance.”
has_name	model_input, model_output, model_param, model_implement, model_hashrate, model_class, model_mature, model_functionality, model_keyword, model_hierarchy	string	“The symbol of the instance.”
has_value	model_input, model_output, model_param	int, float64, string	“The value of the instance.”
has_description	model_input, model_output, model_param, model_hashrate, model_class, model_mature, model_functionality, model_keyword, model_hierarchy	string	“Description of the instance.”
has_timestamp	model_input, model_output	string	“The timestamp of the instance creation.”
code_language	model_implement	string	“The kind of programming language.”
code_description	model_implement	string	“The code of the model.”
hashrate_result	model_hashrate	string, float, int	“The result after running the code.”
hashrate_api	model_hashrate	string	“The API of the computing service.”
preorder_id	model_preorder	int	“The unique identity of model, which is inherited by this model.”
backorder_id	model_backorder	int	“The unique identity of model, which inherits this model.”
similar_id	similar_model	int	“The unique identity of model, whose functionality is similar to this model.”

has_id: 20221106

has_name: ”zinc ion leaching rate”

code_language: ”Python”

code_description: ”import owlready2...”

Industrial manufacturing, especially metallurgy, is composed of multiple processes. Taking the calculation of zinc ion leaching rate in wet zinc smelting as an example, it is necessary to collect the value of the input variables first. And the

next step is to add a model description and the concrete implementation code to the ontology. Finally, we save the ontology by using the standard RDF/XML format.

The owlready2 package, a third-party extension package for Python, can be used to parse and develop the ontology. In this paper, based on the attribute of the industrial model and the data we collect from the manufacturing system, we write scripts to automatically generate instances by using the owlready2 package.

4 Application of the Ontology

In the Internet of Things environment, due to the limitations of size, IOT devices are often equipped with slow processors and limited memory storage. Therefore, most devices can't execute local computing. Computing technology has accelerated the process of the industrial revolution, effectively separating data collection from data computing tasks, and breaking through the limitation of the computing power of industrial equipment. Ontology technology makes industrial models machine-readable and universal. Therefore, integrating distributed computing and ontology technology can provide a platform for data process and exchange between multi-structure systems. Figure 2 shows the architecture of the ontology-based distributed computing application. Firstly, we define the actors involved in the architecture we proposed. Secondly, we demonstrate the working flow of distributed computing. Finally, the experiment result is shown to prove that the ontology we design is feasible and effective.

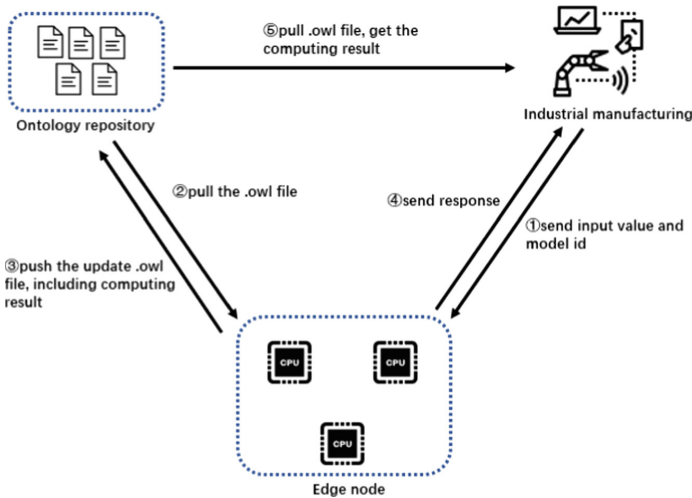


Fig. 2. The architecture of the distributed computing application.

4.1 Actors

The distributed computing system involves three entities. Some detailed descriptions of these entities are listed as followed:

1. Industrial manufacturing: Industrial manufacturing includes the equipment in the factory, e.g. temperature sensor, control center, and assembly line. These devices can collect data generated in the production environment and make requests to nodes in the edge network.

2. Ontology repository: The ontology repository manages the ontologies created by the users. Based on the model id, every owl file is named as a standard format, which can improve the retrieval efficiency significantly.

3. Edge node: Due to the low storage capacity and computing power of industrial devices, edge computing technology has received considerable attention from a large number of researchers. The edge nodes in this system represent the servers close to industrial equipment, which can provide a large number of computing services.

4.2 Distribute Computing Working Flow

The distributed computing process consists of the following steps:

1. Send request: In the first step, the industrial devices collect data, e.g. the reaction temperature, ion concentration, etc. All of these data will be encoded in JSON format, which is a common format for data propagation in networks. Based on the HTTP protocol, the IOT devices broadcast their request package and wait for the response package.

2. Pull the *.owl file: If the edge nodes capture the request package successfully, they will pull the *.owl file from the ontology repository and get the implemented code. According to the implementing language, they run corresponding dockers for supporting multiple programming language environments.

3. Push the update *.owl file: In this step, based on the PBFT (Practical Byzantine Fault Tolerance) algorithm, these edge nodes reach a consensus before they update the computing result. The result will be stored in the data layer and the leader node is responsible for pushing the new *.owl file to the ontology repository.

4. Send response: After the leader, node update the *.owl file, it packages the filename and sends the response package to clients.

5. Get the computing result: The client waits for the response package until it catches the file name, and pulls the *.owl file from the ontology repository. Parsing the *.owl file by the owlready2 python package, the client can fetch the computing result from the data layer.

5 Conclusion

The paper presents the development of computational ontology with a particular focus on encoding machine-readable knowledge of industrial models. Meanwhile,

we constitute a prototype distributed computing application that is used to facilitate the data computing of the manufacturing process. Reviewing existing ontologies, it is clear that all of them are suffering problems such as developing with different levels of granularity, lacking interoperability, and redundant definitions of the same entities. The strategy we proposed provides a standardized development of ontology-based industrial models in an attempt to optimize interoperability between industrial models.

The data collected from heterogeneous resources may involve with the privacy of industrial manufacturing. If the data transmission is intercepted by hackers, the industrial production system has to face large-scale network paralysis. Therefore, enhancing the security of data and adding access control is the future work we will focus on.

References

1. Wei, H.L., Mukherjee, T., Zhang, W., Zuback, J.S., Knapp, G.L., De, A., DebRoy, T.: Mechanistic models for additive manufacturing of metallic components. *Progress Mater. Sci.* **116**, 100703(2021)
2. Ghahramani, M., Qiao, Y., Zhou, M.C., d O'Hagan, A., Sweeney, J.: AI-based modeling and data-driven evaluation for smart manufacturing processes. *IEEE/CAA J. Automatica Sinica* **7**(4), 1026–1037(2020)
3. Noy, F.N., McGuinness, D.L.: Ontology development 101: a guide to creating your first ontology. Stanford knowledge systems laboratory technical report KSL-01-05 (2001)
4. Järvenpää, E., Siltala, N., Hylli, O., Lanz, M.: The development of an ontology for describing the capabilities of manufacturing resources. *J. Intell. Manuf.* **30**(2), 959–978 (2019)
5. Dinar, M., Rosen, D.W.: A design for additive manufacturing ontology. *J. Comput. Inf. Sci. Eng.* **17**(2) (2017)
6. Saha, S., Li, W.D., Usman, Z., Shah, N.: Core manufacturing ontology to model manufacturing operations and sequencing knowledge. *Service Oriented Computing and Applications*, 1–13 (2023)
7. Sanfilippo, E.M., Belkadi, F., Bernard, A.: Ontology-based knowledge representation for additive manufacturing. *Comput. Ind.* **109**, 182–194 (2019)
8. Lemaignan, S., Siadat, A., Dantan, J.Y., Semenenko, A.: MASON: a proposal for an ontology of manufacturing domain. In: *IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications (DIS'06)*, pp. 195–200 (2006)
9. Elkan, C., Greiner, R.: Building large knowledge-based systems: representation and inference in the cyc project: DB Lenat and RV Guha. *Artificial Intelligence* (1993)
10. Mayer, R.J.: Information integration for concurrent engineering (IICE). In: *IDEF3 Process description capture method report* (1995)
11. López, M.F., Gómez, P.A., Sierra, J.P., Sierra, A.P.: Building a chemical ontology using methontology and the ontology design environment. *IEEE Intell. Syst. Appl.* **14**, 37–46 (1999)
12. Farazi, F.: OntoKin: an ontology for chemical kinetic reaction mechanisms. *J. Chem. Inf. Model.* **60**(1), 108–120 (2019)

13. Tim, B.L., Hendler, J., Lassila, O.: The semantic web. *Sci. Am.* **284**(5), 34–43 (2001)
14. Bitsch, G., Senjic, P., Askin, J.: Dynamic adaption in cyber-physical production systems based on ontologies. *Procedia Comput. Sci.* **200**, 577–584 (2022)
15. Zhao, Y.Y., Liu, Q., Xu, W.J., Yuan, H.Q., Lou, P.: An ontology self-learning approach for CNC machine capability information integration and representation in cloud manufacturing. *J. Ind. Inf. Integr.* 25, 100300 (2022)