IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received August 30, 2021, accepted September 21, 2021, date of publication September 23, 2021, date of current version October 4, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3115165

Building SysML Model Graph to Support the System Model Reuse

CHAO FU, JIHONG LIU, AND SHUDE WANG
School of Mechanical Engineering and Automation, Beihang University, Beijing 100191, China

Corresponding author: Jihong Liu (ryukeiko@buaa.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB1701702, and in part by the Basic Research Program for National Defense of China under Grant JCKY2018204B010.

ABSTRACT With the implementation and application of MBSE (Model Based System Engineering) in enterprises, a vast number of system model files is constantly and increasingly generated. The problem of how to acquire the knowledge carried by the system model and reuse the system model is urgently required to be solved. Therefore, a cross-products system model graph construction method supporting system model reuse is proposed. System model graph is constructed by defining the SysML metamodel ontology and the construction strategy is given. The entity alignment based on the relationship set and the rule-based reasoning is applied to achieve the fusion of multiple subgraphs, and the semantic extended retrieval of design requirements is realized based on this graph. Finally, the case studies of the Automatic Climate Control Integrated System of car and satellite design verify the effectiveness and practicability of the method, supporting system model reuse. In conclusion, the paper indicates that it is feasible to express the system model with knowledge graph and the constructed system model knowledge graph can well support the retrieval and reuse of cross- products system models.

INDEX TERMS Model based system engineering, SysML, system model reuse, knowledge graph.

I. INTRODUCTION

With the extension and development of system engineering, gradual increase in customer demand for multidisciplinary and complex products, the complexity of products is also increasing. Because of its integration of hardware (electronics, machinery) and software components, there is a coupling between the various disciplines, resulting in the cost and difficulty of product development. When solving the complexity problem between different fields, traditional Document Based System Engineering (TBSE) is difficult to meet the current research and development needs, and Model-based System Engineering (MBSE) is the best choice.

MBSE has been applied in aerospace, vehicle, shipbuilding and other manufacturing enterprises at present. MBSE is a model-centric approach to different disciplines (including mechanical, electrical and software).It is expected to replace the previous document-centric approach by being integrated into the systems engineering process to change future systems engineering practices [1]. MBSE supports the life cycle of system engineering activities through formal modeling [2], with system complexity control and management capabilities, information consistency, and global traceability.

Knowledge reuse is considered to be the key element to support agile and effective decision-making process in product development process [3]. With the application of MBSE in enterprises, system models of the same type product or a series of products produced by the same enterprise have gradually accumulating. In order to improve the efficiency and quality of product design, it is very significant to make use of the knowledge acquired from those system models [4]. In other words, model reuse is an important product of the application of knowledge from system models. When a series of existing system models are available, it is obvious that it requires much less effort to design a system variant by modifying a set of requirements of those system models, compared with designing a totally new one. In fact, most enterprises are doing the innovative design of new products based on the design data of existing products. Those accumulated system models are the precious wealth of enterprises. System model reuse refers to the process of quickly reusing the built system model and the knowledge carried by the model through a specific method or framework.

The associate editor coordinating the review of this manuscript and approving it for publication was Bing Li.

Since the reuse of system model is necessary, how to expediently find the information and knowledge carried by the system model that designers much need? Some researchers have begun to focus on system model reuse. In summary, it can be divided into three categories:

(1) Framework-based System model reuse: Kerzhner *et al.* [5] proposed the SysML based engineering analysis model reuse framework (MasCoMs), based on which the designer can complete the system modelling and improved the reusability of the model. Shinozaki *et al.* [6] research on mechanical products modelled by RFLP-based framework, through requirements trace, designers can complete new designs by changing requirements and improves the reusability of models. In addition, the SysML modelling Software manufacturers made modelling methodology into modelling framework to support model reuse, such as the customized Magicgrid methodology of MagicDraw [7] and harmony-SE of IBM.

(2) Retrieval-based System model reuse: Morillo *et al.* [8] used the UML model represented by XML and studied the model retrieval based on RSHP (Relationship). Mendieta *et al.* [4] applied this idea to the SysML model retrieval process. By transforming the model elements and model relationships into products and relationships in the RSHP model, the RSHP model was used to support the retrieval capability to complete the retrieval of system models. Tian [9] used multi-physics information model (MIM) to integrate design and simulation knowledge in different fields, and extracted the required knowledge from MIM through intelligent modelling system to instantiate sub model, and Multiphysics simulation models can be built by sub models. The ability of major SysML modelling software to support search reuse is also limited. For example, MagicDraw does not support natural language retrieval of models. Papyrus and Rhapsody need to use regular expressions to improve the difficulty of searching. And these tools can only provide internal retrieval of the currently open SysML file. They have not established a model library, which cannot provide global retrieval function within multiple model files. This kind of retrieval method only provides a relation-based retrieval mechanism, and it is difficult to support the designer's multi-faceted retrieval requirements.

(3) Pattern based System model reuse: Wu *et al.* [10] discussed and analysed the reuse of proprietary using patterns in model based system engineering, which integrates the capitalization, reuse and update capabilities of patterns in the form of library. Yuan *et al.* [11] focused on the trade-off process of design, and achieved the reuse of the model by defining the way of solving the problem model. Each mode includes basic information, problem description, solution, impact and other attributes. By this method semantically matching the input questions and finding the solution to the problem. Although this method implements an automated reuse process, it is difficult to generalize to the reuse of all design models.

In summary, framework-based system model reuse and pattern based System model reuse requires SysML modelling in a specific modelling framework or established design pattern [12], which limits the way the model is created. While, retrieval-based system model reuse can effectively apply the constructed system model, no matter which methodology this system model is based on. However, current retrieval-based system model reuse researches only focus on how to retrieve and reuse a particular product's system model. In fact, different products, of the same type, are often share the similar or even the same functions and structures, so cross-products system model retrieve and reuse is necessary. Besides, all of the above researches need to rely on the SysML language structure itself, and only support specific levels of model reuse (such as framework level reuse or element level reuse). But, it is difficult to reuse the system model organized by SysML. Because the nine diagrams of SysML are nested to express the composition and function of a system. The framework level model reuse will miss a lot of necessary design details, and element model level model reuse provides just useful design information without context, but not design knowledge [13]. As we know, some technologies focusing on product reuse, such as product line engineering (PLE) and product family technology are one of the systematic methods to realize large-scale product reuse. Trujillo *et al.* [14] proposes to use SysML to model the variability of products to deal with the variability and change ability required by customization, and applies this method to the development of product line for power generation of wind turbine system. Hummell and Hause [15] puts forward the concepts and methods of model based product line engineering. For the system model, it is important to reuse design modules and design elements from different levels, which is not clearly given in these researches.

So does SysML model have a better organization form to facilitate the reuse of system model? The answer of this paper is to use knowledge graph technology to support the reuse of system model. It's a new and challenging idea. Then why knowledge graph? On the one hand, SysML is a graphical modelling language, which is oriented to model information description. In fact, the graphical modelling language defines the modelling elements and the connecting lines between them, which can be transformed into the entities in the graph and the relationships between them. Through the transformation, the semantic networks composed of various design elements in the system model are actually constructed, which is useful for knowledge mining and knowledge reuse. On the other hand, the system model graph is easier to be processed by computer, and the entity nodes of the graph retain the original relationship of the design model, Therefore, we can mine the knowledge that designers need from these interrelated nodes. Besides, there is no relationship among system models of different products. Therefore, it is necessary to find an organization that not only does not destroy the internal relationship of the system model, but also can express the relationship among system models of different products. The knowledge graph uses the triple expression of subject, relation and object to describe the connection between entities [16]. Through the way transforming the system model into a knowledge graph, each entity node in the graph is linked to the corresponding system model. Thus, various types of reuse such as retrieval based on the graph can complete the reuse of the system model.

This paper aims to use knowledge graph technology to build and fuse system model graphs of different products, so as to establish the relationship among system models of different products and realize the retrieval of the useful system models. The system model graph is helpful to analysis/integrate between cross-domain models, which enables decision makers to evaluate the quality and suitability of the models for reuse purposes. While, how to use these retrieved system models for a new design is not included in this study. The research of this paper can expand and improve the MBSE theoretical system, and also provide an effective method and new way for the system design of complex products. The main contribution and innovations of this paper are as follows:

(1) A method of using knowledge graph as the organization form of SysML model to facilitate the reuse of system model is proposed. The reuse level and reuse framework of system model is analysed, and the reuse framework of system model based on knowledge graph is established.

(2) A top-down system model graph construction and fusion method are proposed. By defining the SysML meta-model ontology as the schema layer of the system model graph, the construction method and process of the system model graph based on this are established. By using the entity alignment on the basis of relation set and the map completion on the basis of rule-based reasoning, the multi graph fusion is realized. So, the system model graph can be used as knowledge base to support system model reuse.

(3) The reuse method of model retrieval is given. Semantic retrieval based on the system model graph can be used to query and locate the system model of different products.

The structure of this paper is set as follows. The second section introduces the related research of system modelling language, MBSE methodology and the basic theory of knowledge graph. And the third section summarizes the main methods of this paper. Then, the third section completes the construction of the system model graph by defining the SysML meta-model ontology and expresses the construction process. Multi-graphs fusion is achieved by entity alignment based on relation set and rule completion based on rule-based reasoning in section 5, and this section introduces a reuse method of system model graph: model retrieval. Finally, the method proposed in this paper is verified by two examples.

II. RELATED WORKS

A. SYSTEM MODELLING LANGUAGE AND MBSE METHODOLOGY

In order to solve the problem of system modelling, the Object Management Group (OMG) and International Council on Systems Engineering (INCOSE) reuse and extend UML 2.0, publish SysML, which become the standard language of MBSE [17], [18]. As a general graphical modelling language, SysML supports the analysis, design and verification of complex systems, including hardware, software, information, personnel, processes and facilities. SysML, which is independent of modelling methods, supports the practice of MBSE [19]. SysML contains 9 diagrams in 4 categories, which respectively describe the system requirements, behaviour, and structure and parameter information.

Modelling method refers to a set of modelled design tasks performed in order to create a model [20]. It is essentially a collection of related processes, methods and tools in the modelling process. Modelling methods span all phases of the system engineering life cycle, but not all phases are needed in every project, so modelling methods need to be customized to meet specific needs. The modelling method guides the development of the system model by providing a roadmap for all members of the R&D team to ensure the consistency, guidance, and fidelity of modelling in the design process. Common MBSE modelling methodology mainly includes Harmony-SE (Harmony Systems Engineering), OOSEM (Object-Oriented Systems Engineering Method), RUP SE (Rational Unified Process for Systems Engineering) and aMBSE (Agile Model-Based Systems Engineering, aMBSE) etc [21].

B. KNOWLEDGE GRAPH

The knowledge graph is essentially a semantic network, in which each node represents an entity in reality and the connections between entities represent the their semantic relationship. The concept of knowledge graph was first proposed by Google in 2012 to improve its search "the world is made up of everything, not a string" [22]. It provides effective technical support for knowledge representation, knowledge mining and knowledge reuse. Therefore, building a system model graph can provide an effective way to achieve system model reuse.

Due to the diversity and complexity of knowledge graph relationships, it is not possible to store them in a general relational database. There are two main forms of storage [23]: RDF files and Graph Database. RDF is a kind of semantic network. It stores the knowledge in the form of triples. Generally, the software analyses the files in RDF format, such as Jena and MarkLogic. On the other hand, the graph database, as a kind of non-relational database (NoSQL), conducts the graph with basic elements such as points, lines, and polygons, and stores them according to a certain topological data structure. It also has a special database language for adding and checking, deleting, changing and other operations, mainly including Neo4j and so on. In comparison, the graph database has advantages in terms of applicability, processing efficiency of big data and operability.

SysML is a graphical modeling language. Its basic nodes are composed of a model, model attribute, model relationship, model location and other information. In order to make the

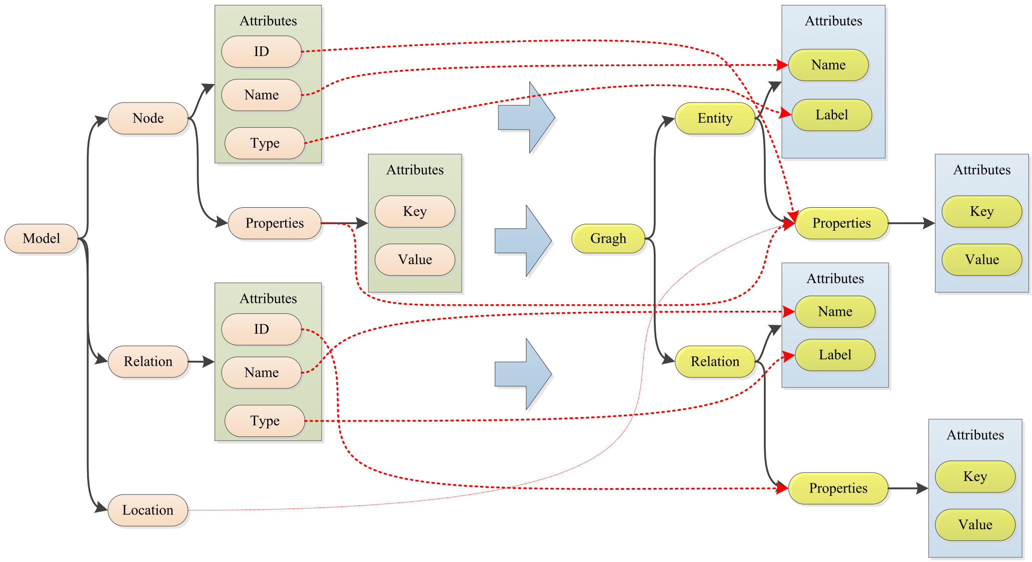

FIGURE 1. The mapping relationship between system model and knowledge graph.

knowledge graph suitable for model reuse and enrich model's relationships, the attributes in the system model are represented by entities to ensure that other types of models in the knowledge graph are standardized and unique. The mapping relationship between system model and knowledge graph is shown in Figure 1.

The branch, decision, merge and other node types in the activity diagram are omitted, and the bolts are connected directly. In retrieval reuse, it is not only the nodes themselves, but also the associated system models.

III. OVERVIEW OF THE APPROACH

A. HIERARCHICAL ANALYSIS OF SYSTEM MODEL REUSE

The expression of the system model is hierarchical. Taking the structural model as an example, the constructed structure is divided into layers including system level, sub-system level and component level. The structural model is mainly constructed and described by the block definition diagram and the internal block diagram. The block definition diagram includes many aspects such as the composition of the structure, interface type, port type, key parameters, signals, constraints, requirements, functions implemented, and behaviour. The way to reuse the system model can be to reuse the complete structural composition of a certain component level or sub-system level, or to reuse the information of a certain underlying component. Therefore, the reuse of system model can not only reuse the high-level information based on the previous design scheme, including requirement analysis structure, function decomposition results, structure composition mode, but also obtain a bottom level model information based on keyword matching, parameter retrieval, function allocation and other ways. It can also realize the unification and standardization of this kind of model by establishing standard interface model, port type, signal type and other models, so as to realize the reuse of this kind of model. The hierarchical analysis of reuse is shown in Table 1.

It is worth noting that both function model and behaviour model exist in the form of activity diagram, so it is necessary to distinguish between function model and behaviour model. In order to distinguish whether the activity describes a function model or a behaviour model, we need to distinguish whether the activity is assigned to a structure. If the activity is assigned to a structure, the process describes the mapping relationship between function and structure, which is the function model. If the activity is not allocated, it is generally considered to describe the behaviour of the design object. In addition, the construction location of activity diagram can also be used as auxiliary information to judge the model type. For example, when the activity diagram is constructed under the structural model, it is the description of the structural behaviour, which is the behaviour model. If the activity diagram is called by a state machine diagram, the activity is also a behaviour model. The function model is composed of its sub functions. For the decomposition relationship of functions, the activity diagram is constructed in the form of action calling other activities, so we use the top-down identification method to obtain the sub function model of the function.

B. PROPOSED FRAMEWORK FOR SYSTEM MODELLING

In fact, there are many ways of model reuse based on system model graph:

1) SYSTEM MODEL RETRIEVAL MATCHING

Although SysML is widely used in the system modeling process, there is very little literature on how to acquire and reuse the knowledge carried in the model. All kinds of model instances in the model base are searched and matched by keyword matching, semantic matching, flow type matching,

TABLE 1. System model reuse hierarchy.

System model	Reuse hierarchy	Reuse method	The role of reuse process
Requirement model	Requirements analysis architecture	Design requirements matching	The architecture framework and traceability relationship including requirements analysis can guide the requirements analysis process of similar designs
	Requirement item	keywords matching	Match similar requirements, find the function and structural models that meet the requirements
Functional model	Total system functions, high-level system functions	Match based on flow type. Retrieval based on stream type	Function decomposition is the key step of conceptual design, including the detailed implementation of function, which can guide the realization of function
	Meta function	Stream type matching. Keyword matching.	Retrieve the behaviour that enables this function. Retrieve the structure with this function
Structural model	High level structure model	Keyword matching. Interface matching.	Reuse of information such as structure composition and internal connection
	Component level structure	Interface based matching. Parameter matching.	Reuse of parameter relation and constraint information of structure
Behaviour model	Various forms of behaviour model	Reuse of structure behaviour model. Function-behaviour-structure mapping	Description of state transition, operation mode and interaction process of structure, etc. The corresponding structure can also be found based on the behavior
Other models	Various standardized models	/	Type selection. Reuse based on correlation model. Interface matching and other methods.

interface type matching and other ways to meet the needs of designers in the design process of rapid design and modeling.

2) PRODUCT CONFIGURATION DESIGN

The proposed system model graph can be used as an infrastructure of Model based product line engineering. A system model graph containing many product design schemes. designers can mine the public components or design elements to support the product line engineering through the technologies such as big data analysis and knowledge discovery. Using the system model graph, designers can find the required component block, or find just a design element (such as an interface), for replication and reuse. The designer can use case-based reasoning and other methods to design and configure new products. The process of product configuration starts from the requirements of customers. By searching and defining a series of rules for the relevant models in the product case library, the product parameters are matched with the needs of customers. Finally, a reasonable feasible scheme is quickly found and the optimal solution is found through configuration evaluation.

3) HETEROGENEOUS SYSTEM MODEL DATA SHARING

The system model files built in different modeling tools are heterogeneous. In order to realize the reuse of heterogeneous model files, the first step is to extract and unify the information and knowledge contained in the model. System model, or semantic network, which stores data in hierarchical structure, has good scalability and readme, and is a common knowledge description language. Through the unified representation and storage of knowledge in different file formats of system model through semantic network, the data sharing of heterogeneous system model can be realized (this paper only developed and verified in Cameo system modeler software).

In this paper, we only study retrieval based model reuse. The proposed framework for system modelling is shown in Figure 2. The process builds a system model ontology with the system model as input, and establishes an index relationship with the model. And reuse is done by searching the system model graph. The method mainly consists of two steps:

Step1: Complete the construction of the system model graph by defining the ontology of the SysML metamodel and establish the construction strategy.

Step2: Multi-graph fusion is achieved by entity alignment based on relation set and graph completion based on rule-based reasoning. Based on the cross-product SysML graph, designer can search the needed system model. The results of searching are sorted and thus a system model that meets the design requirements has been selected to reuse. These two steps are described in detail in sections 4 and 5 below.

IV. SYSTEM MODEL GRAPH CONSTRUCTION BASED ON SysML METAMODEL ONTOLOGY

A. SysML METAMODEL ONTOLOGY CONSTRUCTION

The knowledge graph is divided into two levels, which generally consist of two layers: the pattern layer and the data layer. The pattern layer defines the data pattern and rules of the graph, which is the core level of the knowledge graph, and is usually represented by ontology.

The SysML metamodel ontology refers to the ontology model constructed according to the SysML metamodel, which can guide the automated construction of the system model graph. In order to solve problems of SysML metamodel ontology construction, we need to analyse its concept

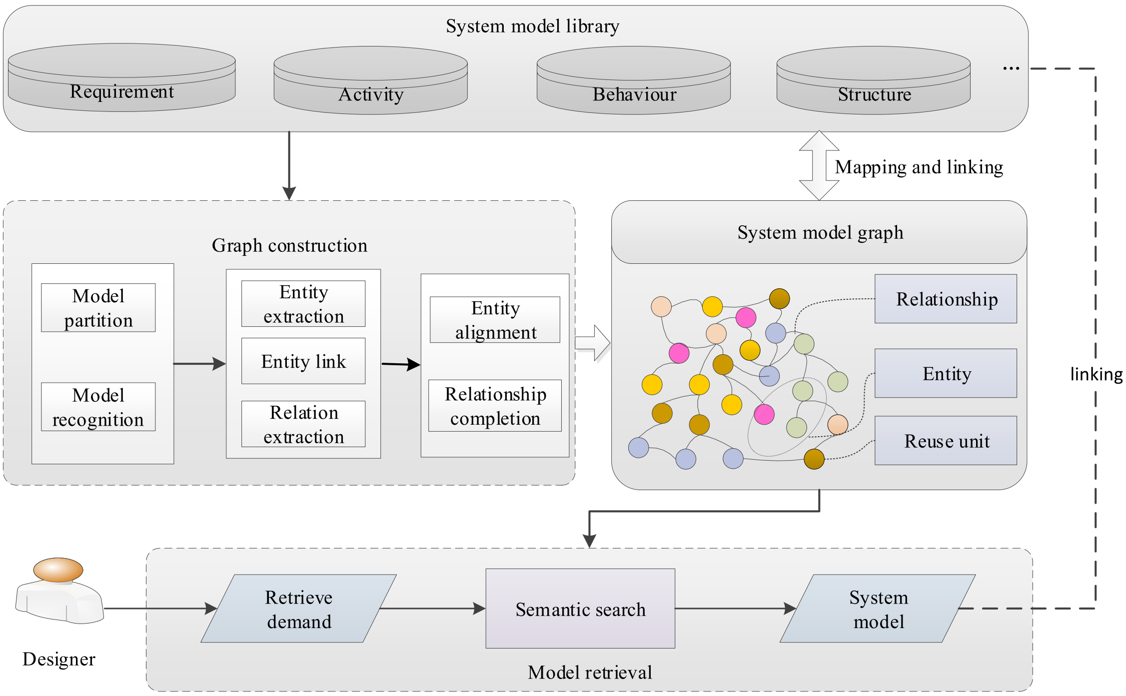

FIGURE 2. Proposed framework for system modeling.

and representation method. The metamodel is an abstract model used to describe the model of the model. This is an important concept in the Meta Object Facility (MOF).

MOF is a meta-model and metadata repository standard proposed by OMG for the purpose of solving data integration and data exchange in different systems. [24]. SysML is extended by UML and follows the MOF specification [25]. The built SysML model is defined by the SysML metamodel. Based on above arguments, the ontology can be derived from the knowledge graph, which acts like the SysML metamodel to the system model. Therefore, the metamodel specification of SysML can be released on the basis of that of OMG, combined with the specific modelling features in the modelling software. Reuse the system model and build the SysML metamodel ontology help support the conversion of the system model to the knowledge graph. This paper analyses the similarities and differences between the meta-model and the concepts and definitions in the ontology, and studies the relationship between the two modelling languages to complete the construction of the meta-model ontology.

The conceptual elements in the ontology are composed by Perez *et al.* by using the taxonomy, and five basic constituent elements [26] are summarized: class or concept, relationship, function (special relationship), axiom and instance. Baclawski *et al.* [27] summarized the conceptual correspondence between UML and OWL. Based on the similar expression between UML and OWL, we can construct the metamodel ontology model based on the SysML metamodel. Although the metamodel is similar to the ontology, there are the following differences:

1. The SysML metamodel and the ontology both apply the inheritance to represent the relationship between the parent class and the child class. But the metamodel follows the UML specification, allowing multiple inheritance subclasses have multiple direct parent classes, such as the ''ConnectableElement'' type whose parent class includes ''ParameterableElement'' and ''TypedElement''. Subclasses (concepts) in the ontology have only one direct parent class (concept).

2. The ontology does not support to establish an inheritance relationship. The relationship type in the metamodel is similar to the definition of the element type, which is defined by inheritance from the parent class. For example, ''PartAssociation'' and ''SharedAssociation'' both inherit the ''Association'' relationship type, and are defined by other attributes such as ''AggregationKind''.

3. The ontology consists of concepts and relationships between concepts. Each concept contains corresponding attributes, which are different from concepts. The partial attributes of the model elements in the metamodel are defined by establishing ''Composition'' with other model elements. That is to say, the attributes in the metamodel cannot be completely mapped to the attributes in the ontology, but the constraint relationship between concepts needs to be established.

4. In addition to the relationship types defined in the metamodel, there are some implicit relationships defined as model attributes. That is, in the metamodel, the definition is made by attributes, and in the ontology, it is converted into a binding relationship. For example, there is a relationship of ''Behavior'' type between ''CallbehaviorAction'' and the corresponding ''Activity''. In the metamodel, it is defined by

the attribute in "Action". In the ontology, it is necessary to construct a constraint relationship between two classes (concepts). The relation table (constraint relation) in metamodel ontology is shown in Table 2 (part).

TABLE 2. Relationship of SysML metamodel ontology.

Relationship type	Acquisition way	Provider	Receiver
Include	Model structure	Element	Element
Allocate	Explicit acquisition	Activity	Block
Behavior	Implicit acquisition	Activity	Action
Association	Explicit acquisition	Block	Block
Connector	Explicit acquisition	Port	Port
Type	Implicit acquisition	Pin	Block
ObjectFlow	Explicit acquisition	InputPin	OutputPin
SyneElement	Implicit acquisition	Pin	ParameterNode
ItemFlow	Explicit acquisition	Port	Port
ObjectFlow	Explicit acquisition	Pin	Pin
Satisfy	Explicit acquisition	Block	Requirement

In order to complete the automatic analysis of ontology model, according to the SysML 1.5 and UML2.0 modeling specifications, this paper uses protégé software to build the metamodel ontology model of SysML. The SysML meta model ontology has 421 concepts and 941 relationships, among which the relationships mainly include "IS-A" (Subclass of) and "Member of". Through the SysML meta-model ontology, we can clearly distinguish the relationship between various model elements, and judge the type of a model. For example, the activity containing Pin and Parameter must be a functional model. As the pattern layer of knowledge graph, SysML meta-model ontology provides the element composition, element hierarchy, element relationship definition and element attribute definition of each system model.

B. SYSTEM MODEL GRAPH CONSTRUCTION PROCESS

By establishing the SysML metamodel ontology, we can obtain the mapping rules of the system model to the knowledge graph transformation, which determines the data source and data pattern of the system model sub-graph. Each system model file contains multiple types of model elements.

For the purpose of model reuse, the model information each type of model needs to acquire is also different. Some implicit model relationships cannot be obtained directly through the modelling software, but need further analysis and clarity. Therefore, in order to accurately and efficiently complete the system model knowledge mapping process, we need to define the system model sub-graph construction process, as shown in Figure 3.

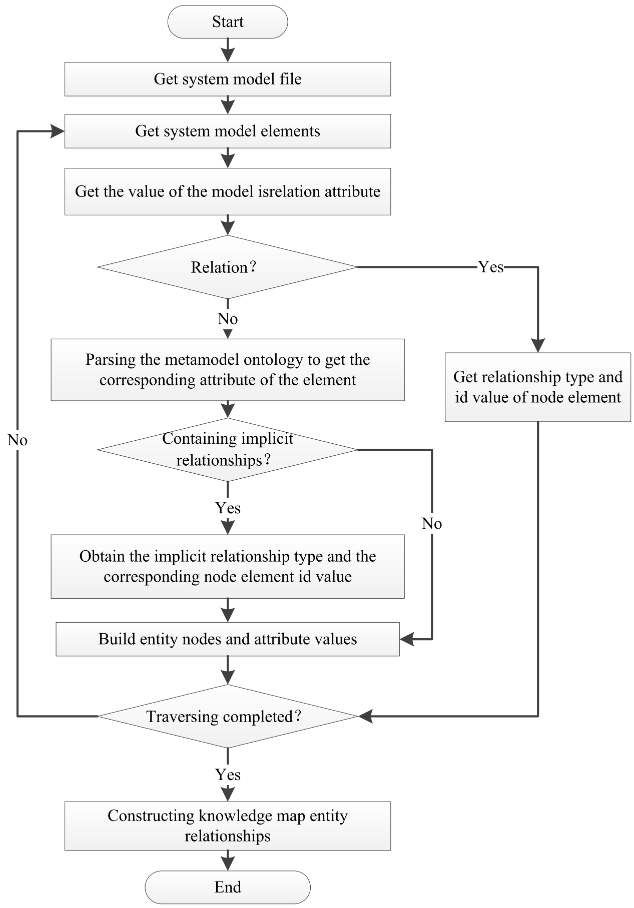

FIGURE 3. System model graph construction process.

As shown in Figure 4, taking a simple system model file (including only functional models) as an example to build system model graph. The activity diagram (function model) mainly includes elements of Action and Parameter node, explicit relationship of Object flow and implicit relationship of CallBehavior. The instance construction process is as follows:

Firstly, the model element can be obtained based on the tree structure of the system model file. By judging whether the element type is a relation, it can be handled in different ways. If the element type is relationship type, such as ObjectFlow, the relationship type of the element and the model element ID value corresponding to the relationship should be obtained. If the element type is activity which is non relational type, then the meta model ontology will be used to obtain the data structure and the corresponding attribute value of the entity node.

Secondly, the non relational model elements should be further processed to determine whether the element contains implicit relationship which is defined by the attributes of model elements. For example, the attribute list of Action model elements contains the attribute type of behavior, and the attribute value type is activity. In system model graph,

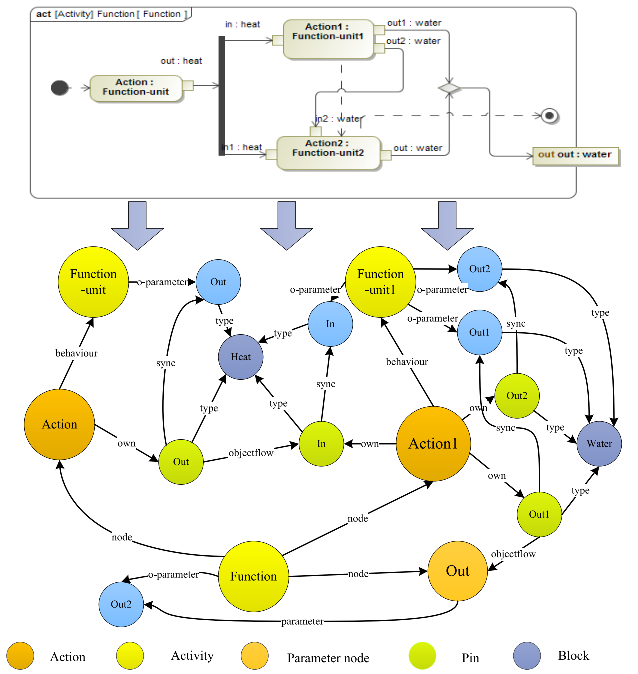

FIGURE 4. An example of construction of system model graph.

we can use the implicit relationship type of CallBehavior to connect the Action and Activity. After all the model elements are traversed, the entity relationship in system model graph can be constructed based on the obtained relationship list. In addition, in order to facilitate the construction and reuse of the system model graph, the node types such as branch, decision and merge in activity diagram are omitted, and Pins are directly connected.

V. MULTI-GRAPH FUSION AND APPLICATION OF SYSTEM MODEL GRAPH

A. ENTITY ALIGNMENT BASED ON RELATIONAL SETS

After converting the system model into sub-graphs, the knowledge graphs converted by different model files are stored in a unified physical location. However, there is no connection relationship among them, and in the specific field or enterprise, the designed system models have a strong correlation. When the built system model is reused based on the knowledge graph, only the original design information already included has been used. It is impossible to reflect the role of knowledge graphs in the fields of knowledge mining and knowledge discovery.

In order to enhance the knowledge discovery ability of the SysML model graph, and to break the translation limits of different model files to obtain the "information island" between entities, it is necessary to amalgamate the obtained knowledge graphs within one, that is, graph fusion. Graph fusion is to combine the entity nodes and entity relationships, representing the same entity in different system model sub-graphs, to form a unified integrated system model graph, namely the system model knowledge base. The process consists of two main steps: physical alignment and graph completion.

Entity Alignment determines whether it points to the same object in the objective physical world by calculating the similarity between entities in different sources of information [28].

If the threshold is met, these entities are represented as the same object, and then can be aligned. At the same time, the information contained in the entity is merged and aggregated together. Its mathematical definition is as in Equation (1), shown at the bottom of the page.

Equation (1) is the logical representation of the entity alignment. The two subgraphs in the system model graph set (G_{set}) are compared. If any entity (e_1) in G_i and any of the entities has the similarity with the threshold, they will be treated as the same entity and merged as one.

The essence of entity alignment is the calculation of entity similarity. It mainly includes entity semantic similarity and structural similarity. That is, the entities own name and the similarity between the attributes and the connection relationship between the entity and other entities.

In the system model graph, different entities need different alignment strategies. The system model mainly consists of five types of models, each of which contains different model elements. Therefore, the alignment strategy is established as shown in Figure 5. In Figure 5, $sim < a, b, c >$ represents the node similarity considering the node characteristics of a, b, c, and sim_0 represents the similarity threshold.

1. The requirement model mainly contains information of the name of the requirement and the description of the requirement. After calculating the similarity between the required entity name and the requirement description attribute, once the threshold is reached, alignment will be performed. When the entities are aligned, the two entities are merged as one and the UIR attributes are added, linked to different system model files.

2. A single functional model transforms multiple entities in the knowledge graph, including functional elements (Activity, Action), input and output parameters (Pin, Parameter), and parameter types. Therefore, when aligning the function model, it is necessary to first align the nodes representing the parameter types. The alignment refers to the strategy 5. Then we need to calculate the similarity of the function nodes that is their string similarity. Finally, if the number and direction of the parameter nodes are the same, the functional model is considered to be the same, and the physical alignment is completed. In the entity fusion process, the corresponding

$$\text{Align}_{\text{entity}}(\text{G}_\text{i},\text{G}_\text{j}) = \{(e_1, e_2, sim)|\forall e_1 \in G_i \wedge \forall e_2 \in G_j, G_i, G_j \in G_{set} \wedge i \neq j, sim \in [0, 1]\} \tag{1}$$

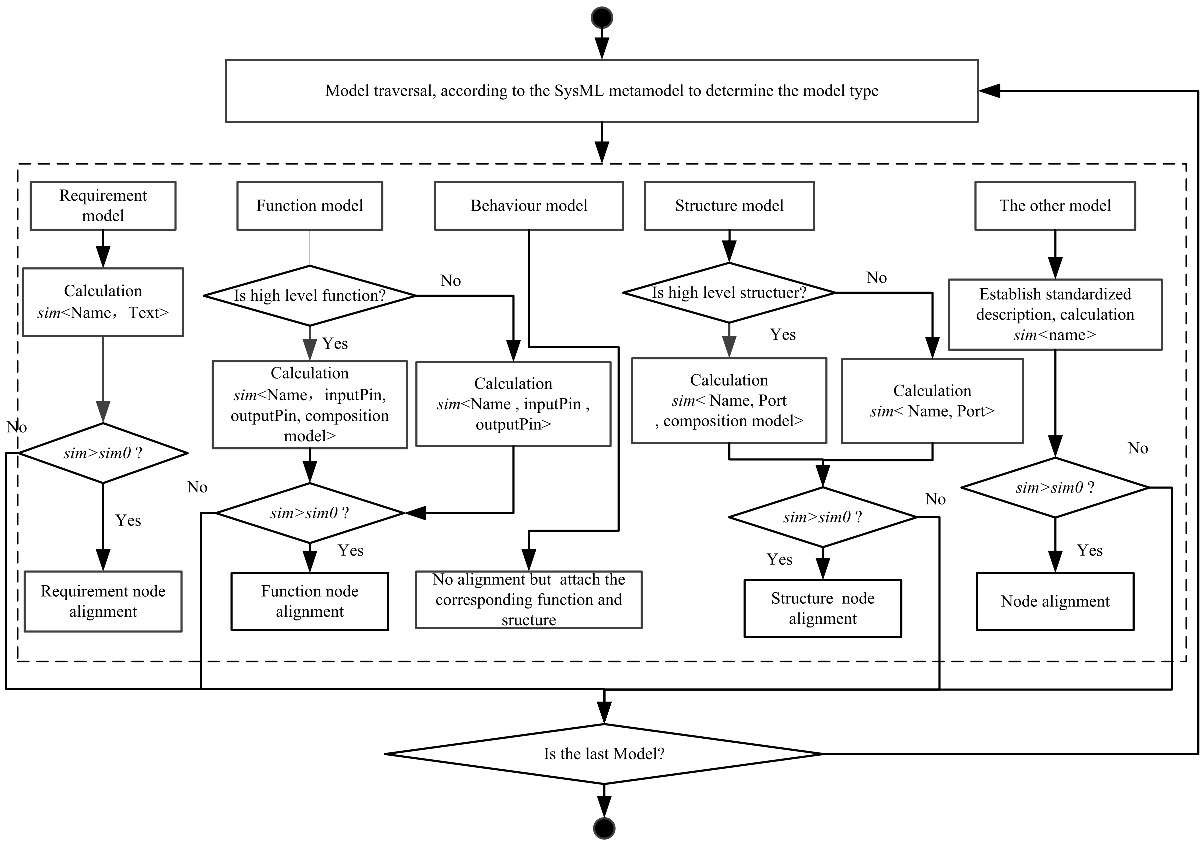

FIGURE 5. Entity alignment strategy.

parameter nodes are merged, and multiple name attributes are added. For feature element nodes, only multiple UIR attributes are added to link to different model files. For high-level functional models (that is, implemented by multiple functional combinations), the functional models they need are also consistent.

3. The alignment of the structural model entities is similar to that of the functional model entities. Structural model entities mainly include structural elements (Blocks), structural components (Parts), interfaces, interfaces (Interface, Port), and entities that contain attributes. Therefore, when aligning the structural model, we need to first align the nodes that represent the interface and port type. The alignment refers to strategy 5. Then calculate the similarity of the interface type, that is, its string similarity. If the number of interfaces and port nodes are the same, the structure is the same and the physical alignment is completed. In the entity fusion process, the corresponding nodes are merged and multiple name attributes are added. For the structure element nodes, only multiple UIR attributes are added to link to different model files. The system, subsystem, and component-level structure also need to be considered for their composition.

4. Because the behavioral model is more complex and the individual model elements contained in it do not have practical meaning, the similarity calculation of the nodes is not performed, but the alignment is completed according to the structure or function described. For example, after two structural models are aligned, the two behavioral models describing the structure will no longer calculate their similarity. In the end, both maintain the Behavior relationship of the entity node, and link to different model files when reused.

5. Other models that need to be standardized ensure their similarity under the same name (name) by standardizing the model library or standard naming conventions. That is, the models with the same name are the same model, and the physical alignment has been completed.

The entity character similarity is calculated by the string similarity based on Edit-Distance. Its similarity calculation equation is as shown in Equation (2) and Equation (3).

$$\mathrm{Ed}\left(s_1, s_2\right) = \frac{\left|\{op_1\}\right|}{\max\left(\mathrm{len}\left(s_1\right), \mathrm{len}\left(s_2\right)\right)} \tag{2}$$

$$\mathrm{SIM}_{st}\left(s_1, s_2\right) = \frac{1}{1 + \mathrm{Ed}\left(s_1, s_2\right)} \tag{3}$$

$\{op_1\}$ is the minimum number of steps required to add, delete, or change characters to modify the string s_1 to s_2, len(s) the number of characters for s, For example, the number of characters in "Dellstatelite" is 13.

B. SYSTEM MODEL GRAPH COMPLETION BASED ON RULE REASONING

After the entity alignment is completed, based on the rules defined in the metamodel ontology, the aligned entity nodes are used as the fulcrum to complement the model graph to improve the integrity of the graph and enhance the knowledge expression and discovery ability. The above mentioned rule reasoning is adding the mined knowledge and conclusions to the knowledge base in the way of rule definition. And based on this, the process of new knowledge reasoning can be completed.

In the process of constructing knowledge graphs, knowledge reasoning has important significance for the mining of new knowledge and the improvement of knowledge base [29]. Through the previous analysis, this study constructs the SysML metamodel ontology. In the ontology, both the hierarchical relationship of the metamodel and the constraint relationship between the metamodels are included. Based on these relationships, rule inference is performed, and the aligned knowledge graphs are complemented to complete the knowledge graph fusion.

The knowledge graph is similar to the instance layer of the ontology, which is equivalent to an instance of the ontology corresponding concept. Instances have attributes and relationships for the corresponding concepts. The rules defined in this paper are mainly the binding relationships between concepts. For example, ''molecular sieve oxygen generator'' (block) can achieve ''allocate'' (Activity). ''Oxygen production function'' reflects (requirement). Then ''molecular sieve oxygen generator'' (block) can satisfy (satisfy) ''requirement for oxygen production''. It should be noted that the defined rules can be used not only in the process of graph fusion, but also in the complement and error correction of a single graph.

The constraint relationship inference rule definition (partial) is shown in Table 3. The rules have the following meanings:

Rule5: indicates the mapping between requirements, use cases, and functions. This rule is similar to Rule1.

TABLE 3. Inference rule definition (partial).

Num	Rule
1	trace(? req,? act) allocate (? act,? block) →satisfy (? req,? block)
2	refine(? req1,? req2) satisfy (? req2,? block) →satisfy (? req1,? block)
3	driveReqt(? req1,? req2) satisfy (? req2,? block) →satisfy (? req1,? block)
4	node(? activity1,? action) behavior(? action,? activity2) →composition(? activity1,? activity2)
5	refine(? req,? use case) behavior(? use case,? act) →trace(? req,? act)

Rule1: represents the mapping between requirements, functions, and structures. That is, if the function reflects the requirement and the structure implements the function, the structure satisfies the function.

Rule2, Rule3: indicates the structure's satisfaction with the requirements. When there is a relationship between refinement and driveReqt between the two requirements, the latter structure is satisfied while satisfying the former. Similarly, the rule also applies to functions.

Rule4: represents the construction of a functional decomposition relationship. The action is used as an intermediary to establish the decomposition relationship between functions.

As shown in Figure 6, node of Activity 1 and node of Activity2 represent two functions. If the two nodes are aligned in the process of graph fusion, two new relationships can be complemented by rule 3. The new relationships are ''Block2 satisfy Req1'' and ''Block satisfies Req2''. The new relationship obtained through rule inference will be automatically built in the system model graph to complete the completion of the graph. The completion method can also be applied to the dynamic evolution and update process of the system model diagram.

C. APPLICATION OF SYSTEM MODEL GRAPH: MODEL RETRIEVAL BASED ON SEMANTIC EXTENSION

In order to solve the problem of poor support for natural language retrieval in current modeling software, this chapter will apply domain ontology to complete the semantic expansion of query vocabulary. The retrieval process is shown in Figure 7. Firstly, the retrieval requirements are segmented. Then, based on domain ontology, semantic extension of keywords is implemented. The new semantic extended set is mapped in the knowledge graph, and the mapping entity set with their URL is pushed to the designer.

The research content of this paper is mainly for multidisciplinary and complex products, which have high complexity and coupling between disciplines. This research has respectively constructed multidisciplinary fields such as machinery, electronics, hydraulics, software, etc. Here, the ontology has a relationship between more than 1,700 domain concepts and more than 7,500 domain concepts.

1) SEGMENTATION AND WEIGHT

In order to ensure the adequacy and accuracy of the search results, we must first classify the search requirements of the designer, and take into consideration the different degrees of importance of different parts of speech in the search statement. According to the results of the part of speech analysis, the central vocabulary in the short sentence can be extracted and given a higher weight, while the attributive farther from the central word has a lower weight. The weight w distribution formula is Equation (4).

$$w_i = \frac{\text{length}(i)}{d_i} / (\sum_{i=0}^{n} \frac{\text{length}(i)}{d_i}) \quad (4)$$

where n is the number of words obtained by the word segmentation, d_i is the distance of the i-th vocabulary from the central word, length (i) is the number of words contained in the vocabulary. If the matched entity or ontology has multiple matching words with the retrieved short sentence, then the accumulation is performed.

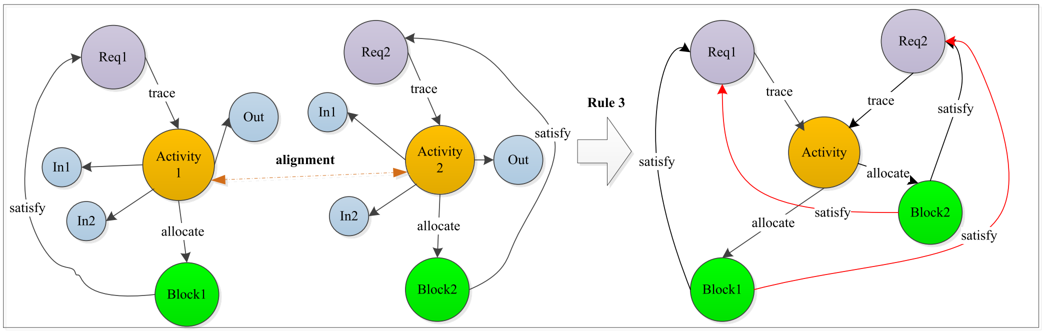

FIGURE 6. Example of relationships completion based on rule reasoning.

2) ONTOLOGY STRUCTURE EXTENSION

In the domain ontology, the closer the distance between the concepts in the hierarchy dimension is, the higher the similarity between concepts is gained. Therefore, the calculation method for defining the similarity between the concept retrieved by the structural dimension extension and its original concept is as shown in Equation (5).

$$\mathrm{sim}_{st}\left(C_i, C_j\right) = \frac{2\,\mathrm{depth}\,(C_0)}{\mathrm{depth}\,(C_i) + \mathrm{depth}\left(C_j\right)} \tag{5}$$

C_i and C_j are two similar concepts in domain ontology, depth(C) is the depth of C. C_0 is the largest common parent of the depths of C_i and C_j. This formula indicates that the greater the depth of the node is, the higher the similarity between the node and the neighbouring node is gained. This method is the same as the domain ontology subdivision.

3) CONSTRAINT SEMANTIC EXTENSION

The domain ontology contains multiple constraint relationships. To facilitate the calculation, this study only divides the constraint relationship into two categories: synonym relations and other relationships. Among them, the expansion coefficient α of the synonym relationship is 1, and the expansion coefficient α of other relationships is recorded as 0.9. The formula is as shown in Equation (6).

$$\mathrm{sim}_{ct}\,(C_i, C_{i+1}) = \begin{cases} 1, & r\,(C_i, C_{i+1}) = \text{“}synonym\text{''} \\ 0.7, & r\,(C_i, C_{i+1}) = else \end{cases} \tag{6}$$

The concept of constrained semantic extension can continue to extend the structure semantics. In this study, the extended threshold is taken as 0.7. When the obtained similarity between the concept after the extension and the original concept is OSM < 0.7, the semantic extension ends.

The OSM algorithm is like Equation (7):

$$\mathrm{OSM}\,(C_0, C_n) = \prod_{i=0}^{a} \mathrm{sim}_{st}\left(C_i, C_j\right) \prod_{i=0}^{a} \mathrm{sim}_{ct}\,(C_i, C_{i+1}) \tag{7}$$

sim_{st} is the structural semantic similarity, sim_{ct} is the constraint semantic similarity, and a is the constraint dimension expansion number. After the constraint dimension expansion, the structural dimension similarity is the product of the two-segment structure similarity.

4) ENTITY MATCH

The keywords obtained in the previous step are subjected to fuzzy search in the knowledge graph, and the entity names are matched by regular expressions. Based on the extension coefficient β of the relation type in the metamodel ontology, the entity extension is performed in the knowledge graph.

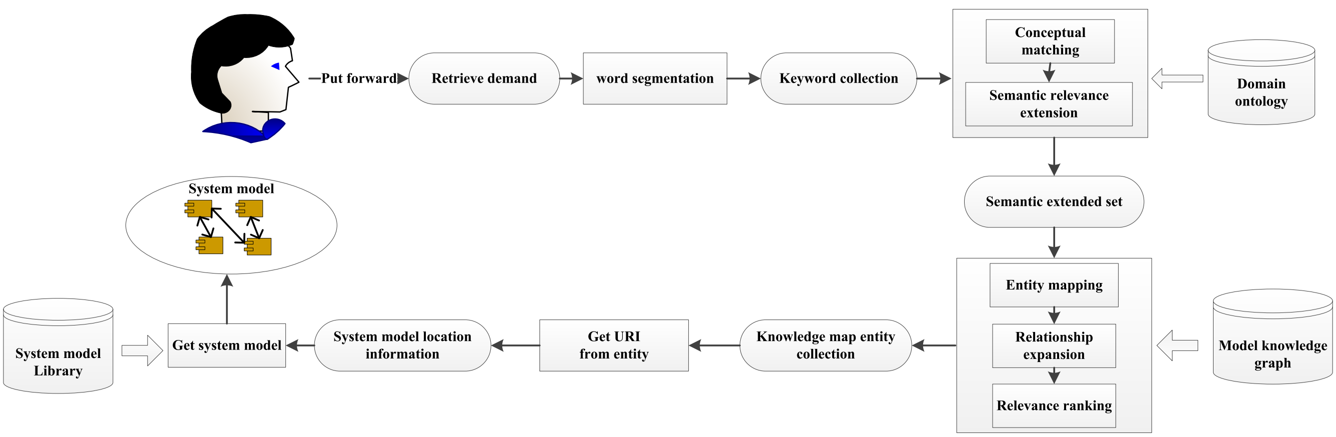

FIGURE 7. Semantic extended retrieval process.

The entity relationship similarity ESM algorithm formula is as shown in Equation (8).

$$\mathrm{ESM} = \prod_{i=0}^{n} \mathrm{sim}_{er}\left(E_i, E_{i+1}\right) \tag{8}$$

The ESM threshold in this paper is also taken as 0.7, which is lower than the end of the entity extension.

The overall similarity TSM algorithm formula is as shown in Equation (9).

$$\mathrm{TSM}_i = \sum_{i=1}^{n} \left(w_i \cdot \mathrm{OSM}_i \cdot \mathrm{ESM}_i\right) \tag{9}$$

where n is the number of words the entity contains. After TSM is obtained, the extended entities are sorted for the designer to select.

VI. DEMONSTRATION

Based on the above research, this section will develop the functional architecture design and system implementation of system model graph construction, system model graph fusion and system model semantic retrieval. The effectiveness of the method was verified by two example.

A. PROTOTYPE SYSTEM

The system model reuses the prototype system with the system model as input, automatically constructs the corresponding system model graph and completes the graph fusion. The system is based on Cameo System Modeler (CSM) system modelling software and is developed in Java language. The Neo4j graph database is used to complete the construction of the knowledge graph, and the ontology visual modelling tool protégé is used to complete the construction of the SysML metamodel ontology. The system architecture is shown in Figure 8.

B. AUTOMATIC CLIMATE CONTROL INTEGRATED SYSTEM OF CAR

The Automatic Climate Control Integrated System of car (ACCIS) is designed for the requirements of the interior climate control in the three proofing environment. Its structure mainly includes A/C Compressor, internal circulation fan, ventilation fan, condenser and other components.

In order to complete the system design, the designer analyses, designs and constructs the requirement model, function model, structure and behaviour model, etc…On the basis of the proposed method, the system model graph is constructed so as to complete the design reuse. The comparison between the number of models in the system model and the number of entity nodes in the knowledge graph is shown in Table 4. This table contains the requirement model, function model and other model types, as well as the corresponding entity nodes.

TABLE 4. Comparison of system models and entity nodes of knowledge graph.

System models		Knowledge graph	
type	number	type	number
requirement model	84	Requirement	84
function model	108	Activity	118
behaviour model	10		
structure model	46	Block	62
item type	16		
Interface model	93	FlowPort	93
data type	20	ValueType	20

According to the table 4, the system model can be successfully converted into nodes and relationships of system model graph based on the proposed method. This shows that it is feasible to express the system model with knowledge graph. The property value of entity node can be used to distinguish different models (such as Block and Activity) of the same type of nodes.

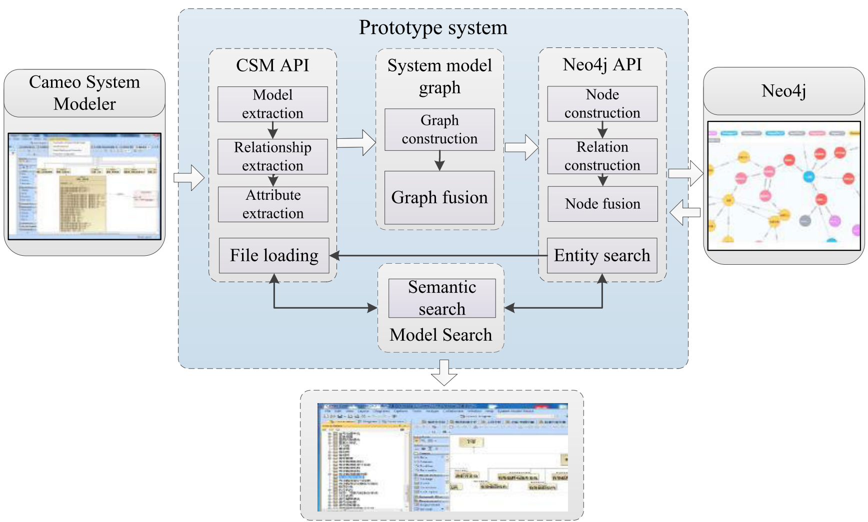

FIGURE 8. Schematic figure of the prototype system.

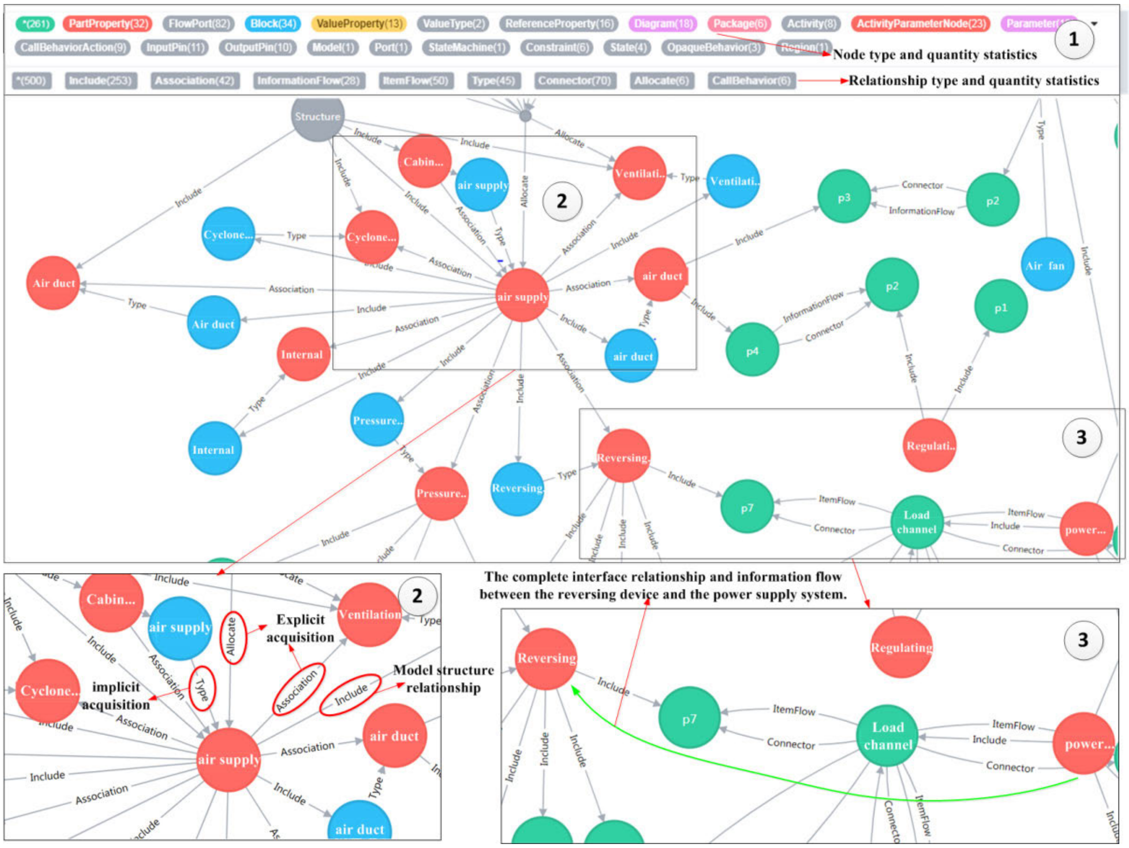

FIGURE 9. System model graph of ACCIS (part).

The system model file of another Automatic Climate Control Integrated System designed for high altitude and cold fighting environment is also converted into the system model graph. Because the two products are similar, their system model graphs can be integrated into a large knowledge graph of Automatic Climate Control Integrated System. The integrated system model graph is shown in Figure 9.

As shown in Fig.9, panel 1 is the type and quantity statistics of nodes and relationship. Panel 2 shows different types of relationships which include implicit acquisition relationships, explicit acquisition relationships, and model structure relationship. How to obtain these relationships is shown in table 3. Panel 3 shows the complete interface relationship and information flow between the reversing device and the power supply system.

Table 5 is the relevant data of fusion of the two system model graphs by using the method mentioned in 5.1 and 5.2. In addition, accuracy, precision, recall and F-measure are also given.

In the table, N_{Total}, N_{needs}, N_{Actual} and N_{Mis} respectively represents the total number of entity nodes, the number of nodes that should be aligned, the number of nodes that are actually aligned, the number of misalignments. It can be seen from the table that most of the entity nodes can be correctly aligned. Those wrongly aligned are mainly caused by the different expression habits of different designers when system modelling.

TABLE 5. The data of knowledge graph fusion.

N_{Total}	N_{needs}	N_{Actual}	N_{Mis}	accuracy (%)	precision (%)	recall (%)	F
261	72	68	3	95.8	95.6	90.3	0.929

Take the block of relief valve structure model as an example, which includes the material flow ports whose type is air and direction is input/output; the power port whose type is DC24V and direction is input/output; the control information port whose type is analog signal interface and direction is input/output. When aligning entities, the model is converted to a set of port direction $P_1 = \{in, out, inout, inout\}$ and a set of port type $P_2 = \{\text{air, air, DC24V, analog, signal}\}$. First, match the model type and name in the system model graph. For the entity nodes whose similarity meets the threshold, the neighbouring nodes of their specific association relationship are also transformed into the above set. Then calculate the similarity according to equation (2) and (3), and align them when the value is 1. Therefore, when the name expression of two entities with the same function, is different, the fusion error will be caused.

C. SATELLITE DESIGN

1) CONSTRUCTION AND INTEGRATION OF SYSTEM MODEL GRAPH

In this case, four satellite design cases are applied to build corresponding system model graph, which are "L-satellite",

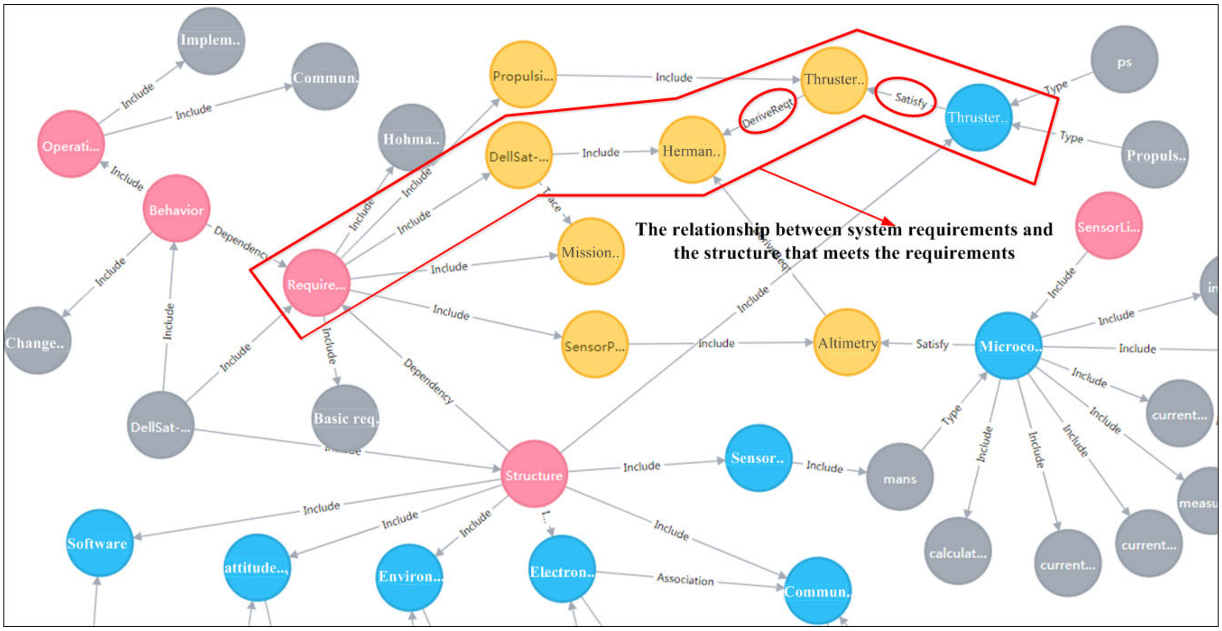

FIGURE 10. System model graph of satellite.

TABLE 6. The data of knowledge graph fusion.

N_{Total}	N_{needs}	N_{Actual}	N_{Mis}	accuracy (%)	precision (%)	recall (%)	F
642	136	130	7	96.9	94.6	90.4	0.925

"Ka-satellite", "DellSat 77" [16] and "CubeSat" [30]. The models contained in these system model files can be transformed into entities and relationships in the system model graph. In the process of transformation, the system will be initiated that if the similarity threshold is met.

Taking the "5-onboard terminal communication uplink" function model as an example, it contains two input and output parameters, i.e. "in: ground" and "out: main satellite", both of which are of type "data". The similarity between the function nodes of "onboard terminal communication uplink" in the system model graph meets the threshold value. After that it indicates to complete the system model graph fusion. In this process, the system will prompt the number of new entity nodes, alignment nodes, new relationships and reasoning completion coefficient. The fusional system model graph obtained is shown in Figure 10.

As shown in Figure 10, the system model graph can show the complete traceability and satisfaction relationship of satellite requirements. The Satellite-77 System Requirements Specification, included in the package of original requirements on one hand, contain the Hohmann Transfer requirement on the other. The Thruster Burn requirement is derived from the Hohmann Transfer requirement, and the block of Thruster Subsystem can satisfy the Thruster Burn requirement. These nodes with detailed relationships are of great significance for future retrieval reuse. In fact, the node relationships constructed by the system model graph can represent the traceability of system requirements and design elements. It can quickly review and analyse the relationship between the elements of the whole model.

TABLE 7. Entity extension results.

Entity name	Type	N_{case}	Score
payload calibrator	Block	4	1
power of payload calibrator	Block	4	0.875
Payload calibration platform	Block	4	0.753
Payload calibration mechanism	Block	4	0.697
Payload calibration function	Activity	4	0.675
Payload measurement equipment	Block	2	0.562
Payload measurement function	Activity	2	0.543
Payload carrier	Block	4	0.423
Load requirement	Requirement	4	0.365
Power subsystem	Block	4	0.307

Table 6 is the relevant data of fusion of the 4 system model graphs by using the proposed method. In addition, accuracy, precision, recall and F-measure are also given. The pre-aligned model elements and the corresponding entity nodes in the graph can be viewed, deleted and modified by the designer.

2) SEMANTIC RETRIEVAL

According to the system model graph of the satellite system, the keywords of "payload calibrator" are searched. The corresponding searching results and scores are obtained, including word segmentation, ontology search, entity search and model selection, as shown in Figure 11.

Take entity extension as an example, the scores of each entity are calculated according to equation (9). As shown in table 7.

From the table, we can find that the nodes have different types. Each node has linked multiple instances (N_{case}). Each instance has its own URL, through which the model files of different instances can be obtained. So designers can compare and discover more meaningful information according to the

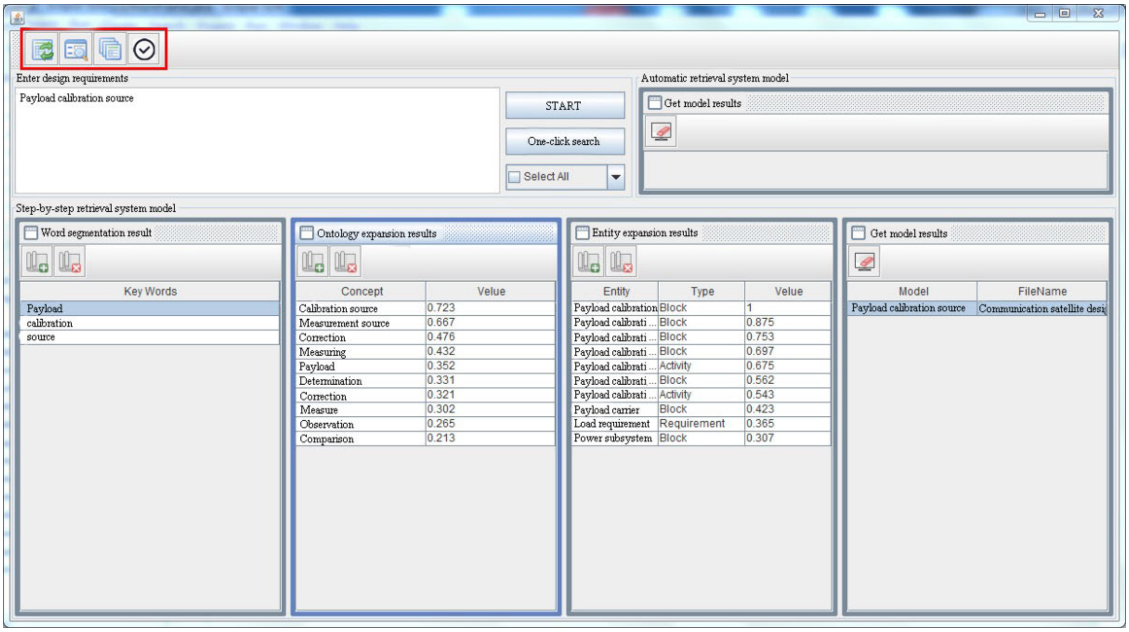

FIGURE 11. Semantic extended retrieval.

instances linked by nodes. When the final note is selected, the corresponding system model file can be loaded in the system modelling software. Designers can use the retrieved system model to design new products or compare these cases to guide the system modelling of new products.

VII. CONCLUSION

This paper completes the construction of the system model graph by defining the SysML metamodel and gives the construction process. In this process, multi-graphs fusion is achieved by entity alignment based on relation set and rule completion based on rule-based reasoning, and the method of reusing the system model with the constructed knowledge graph is semantic extension retrieval. The second development of the CSM software was completed using the Java programming language, and the prototype system development was completed. The main function of the system is verified by a satellite system model reuse process, proving the feasibility and effectiveness of the proposed method.

In general, the main contribution of our approach lies in that it provide a new idea and method to realize the retrieval and reuse of system model across products. The method use knowledge graph technology to build and fuse system model graph of different products. The final case shows that the knowledge graph can effectively express the relationships in SysML model. It can be predicted that when an enterprise builds a complete system model graph for its series of products, the knowledge contained in the system model can be effectively mined and reused, thus enhancing the design efficiency and quality of the enterprise.

In the near future, several issues still need to be further explored to improve the practicability of our approach:

(1) In the process of the system model graph construction, although the ontology layer of the graph is constructed based on the SysML metamodel specification, the interface functions for obtaining model information are based on specific software tools. In the future, it is still necessary to study the construction and fusion of the cross-tool system model graph to truly realize the heterogeneous system model data sharing.

(2) The strength of knowledge graph technology not only consists in the organization of model, but also in the basis of knowledge reasoning. The ultimate aim of system model reuse is that once the designer inputs the design requirements, the system can automatically configure the design scheme to meet the product requirements. Using the system model graph to realize the product configuration design is also our future work.

(3) With the development of artificial intelligence, technologies, such as big data analysis, machine learning and deep learning have become more and more mature. It is a promising idea to introduce artificial intelligence into the field of design. In the future, we can build a more perfect system model graph and use big data analysis and other technologies to mine and discovery useful design knowledge, so as to better support innovative design.

REFERENCES

[1] J. Holt and S. Perry, *SysML for Systems Engineering*. Edison, NJ, USA: IET, 2008, doi: 10.1049/PBPC007E.

[2] J. Gregory, L. Berthoud, T. Tryfonas, A. Prezzavento, and L. Faure, "Investigating the flexibility of the MBSE approach to the biomass mission," *IEEE Trans. Syst., Man, Cybern. Syst.*, early access, Feb. 7, 2020, doi: 10.1109/TSMC.2019.2958757.

[3] R. Krishnan and S. V. Bhada, "An integrated system design and safety framework for model-based safety analysis," *IEEE Access*, vol. 8, pp. 146483–146497, 2020.

[4] R. Mendieta, J. L. de la Vara, J. Llorens, and J. M. Álvarez-Rodríguez, "Towards effective SysML model reuse," in *Proc. 5th Int. Conf. Model-Driven Eng. Softw. Develop.*, 2017, pp. 536–541.

[5] A. A. Kerzhner, J. M. Jobe, and C. J. J. Paredis, "A formal framework for capturing knowledge to transform structural models into analysis models," *J. Simul.*, vol. 5, no. 3, pp. 202–216, Aug. 2011.

[6] M. Shinozaki, F. Mhenni, J.-Y. Choley, and A. Ming, ''Reuse of SysML model to support innovation in mechatronic systems design,'' in *Proc. Annu. IEEE Int. Syst. Conf. (SysCon)*, Apr. 2017, pp. 1–6.
[7] A. Morkevicius, A. Aleksandraviciene, A. Armonas, and G. Fanmuy, ''Towards a common systems engineering methodology to cover a complete system development process,'' in *Proc. INCOSE Int. Symp.*, vol. 30, 2020, pp. 138–152.
[8] J. L. Morillo, J. M. Fuentes, and J. Morato, ''UML retrieval and reuse using XMI,'' in *Proc. Lasted Int. Conf. Softw. Eng. DBLP*, 2004, pp. 740–746.
[9] F. Tian and M. Voskuijl, ''Automated generation of multiphysics simulation models to support multidisciplinary design optimization,'' *Adv. Eng. Informat.*, vol. 29, no. 4, pp. 1110–1125, Oct. 2015.
[10] Q. Wu, D. Gouyon, É. Levrat, and S. Boudau, ''Use of patterns for know-how reuse in a model-based systems engineering framework,'' *IEEE Syst. J.*, vol. 14, no. 4, pp. 4765–4776, Dec. 2020.
[11] W. Yuan, Y. Liu, J. Zhao, and H. Wang, ''Pattern-based integration of system optimization in mechatronic system design,'' *Adv. Eng. Softw.*, vol. 98, pp. 23–37, Aug. 2016.
[12] M. Kejriwal, J. Sequeda, and V. Lopez, ''Knowledge graphs: Construction, management and querying,'' *Semantic Web*, vol. 10, no. 6, pp. 961–962, Oct. 2019.
[13] H. Wang, V. Thomson, and C. Tang, ''Change propagation analysis for system modeling using semantic web technology,'' *Adv. Eng. Inform.*, vol. 35, pp. 17–29, Jan. 2017.
[14] S. Trujillo *et al.*, ''Coping with variability in model-based systems engineering: An experience in green energy,'' in *Proc. Eur. Conf. Modelling Found. Appl.* Berlin, Germany: Springer, 2010, pp. 293–304.
[15] J. Hummell and M. Hause, ''Model-based product line engineering–enabling product families with variants,'' in *Proc. IEEE Aerosp. Conf.*, Mar. 2015, pp. 1–8.
[16] J. Holt and S. Perry, *Foundations for Model-Based Systems Engineering From Patterns to Models*. London, U.K.: IET, 2016.
[17] (2009). *Object Management Group, Systems Modeling Language specification*. [Online]. Available: https://www.omg.org/spec/SysML
[18] *Unified Modeling Language specification*, Object Management Group, Needham, MA, USA, 2007.
[19] A. L. Ramos, J. V. Ferreira, and J. Barcelo, ''Model-based systems engineering: An emerging approach for modern systems,'' *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 1, pp. 101–111, Jan. 2012.
[20] L. Delligatti, *SysML Distilled: A Brief Guide to the Systems Modelling Language*. Reading, MA, USA: Addison-Wesley, 2013.
[21] *A Survey of Model-Based Systems Engineering (MBSE) Methodologies, Rev. B*, Estefan, Seattle, WA, USA, 2013.
[22] J. Pujara, H. Miao, L. Getoor, and W. Cohen, ''Knowledge graph identification,'' in *Proc. Int. Semantic Web Conf.*, 2013, pp. 542–557.
[23] C. Wang, X. Ma, J. Chen, and J. Chen, ''Information extraction and knowledge graph construction from geoscience literature,'' *Comput. Geosci.*, vol. 112, pp. 112–120, Mar. 2018.
[24] T. Wei, ''Meta object facility,'' in *Encyclopedia of Database Systems*. Springer, 2009, doi: 10.1007/978-0-387-39940-9_914.
[25] *Unified Modeling Language Specification*, Object Management Group, Needham, MA, USA, 2007.
[26] A. Gomez-Perez and O. Corcho, ''Ontology languages for the semantic web,'' *IEEE Intell. Syst.*, vol. 17, no. 1, pp. 54–60, Jan. 2002.
[27] K. Baclawski, M. K. Kokar, P. A. Kogut, L. Hart, J. Smith, J. Letkowski, and P. Emery, ''Extending the unified modeling language for ontology development,'' *Softw. Syst. Model.*, vol. 1, no. 2, pp. 142–156, Dec. 2002.
[28] G.-W. You, S.-W. Hwang, Y.-I. Song, L. Jiang, and Z. Nie, ''Efficient entity translation mining: A parallelized graph alignment approach,'' *ACM Trans. Inf. Syst.*, vol. 30, no. 4, pp. 1–23, Nov. 2012.
[29] R. Fagin, J. Halpern, and Y. Moses, *Reasoning About Knowledge*. Cambridge, MA, USA: MIT Press, 2003.
[30] S. C. Spangelo, D. Kaslow, C. Delp, B. Cole, L. Anderson, E. Fosse, B. S. Gilbert, L. Hartman, T. Kahn, and J. Cutler, ''Applying model based systems engineering (MBSE) to a standard CubeSat,'' in *Proc. IEEE Aerosp. Conf.*, Mar. 2012, pp. 1–20.

CHAO FU is currently pursuing the Ph.D. degree with the School of Mechanical Engineering and Automation, Beihang University. His research interests include model-based systems engineering and reliability-based multidisciplinary design optimization.

JIHONG LIU received the Ph.D. degree in mechanical engineering from Tokyo Metropolitan University, Japan, in 1996. He is currently a Professor with the School of Mechanical Engineering and Automation, Beihang University. He has published more than 150 journals and conference papers. His research interests include complex product engineering, knowledge management and knowledge engineering, artificial intelligence in design, and reliability-based multidisciplinary design optimization.

SHUDE WANG received the M.A. degree in mechanical engineering from the School of Mechanical Engineering and Automation, Beihang University. His research interest includes model-based systems engineering.