



A Semantic Ontology-Based Approach to Support Model-Based Systems Engineering Design for an Aircraft Prognostic Health Management System

Jinwei Chen^{1,2}, Yifan Chen¹, Zhenchao Hu¹, Jinzhi Lu^{2*}, Xiaochen Zheng², Huisheng Zhang¹ and Dimitris Kiritsis²

¹The Key Lab. of Power Machinery and Engineering of Education Ministry, School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai, China, ²Institute of Mechanical Engineering, School of Engineering, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland

OPEN ACCESS

Edited by:

Hongbae Jun,
Hongik University, South Korea

Reviewed by:

Ho-Jin Hwang,
Joongbu University, South Korea
Greg Zacharewicz,
Institut Mines -Télécom Mines Alès,
France

*Correspondence:

Jinzhi Lu
jinzhi.lu@epfl.ch

Specialty section:

This article was submitted to
Life Cycle Engineering,
a section of the journal
Frontiers in Manufacturing Technology

Received: 28 February 2022

Accepted: 22 April 2022

Published: 31 May 2022

Citation:

Chen J, Chen Y, Hu Z, Lu J, Zheng X,
Zhang H and Kiritsis D (2022) A
Semantic Ontology-Based Approach
to Support Model-Based Systems
Engineering Design for an Aircraft
Prognostic Health
Management System.
Front. Manuf. Technol. 2:886518.
doi: 10.3389/fmtec.2022.886518

The Prognostic and Health Management (PHM) system of an aircraft has complex structures and diverse functions. It is highly coupled with other systems, such as the avionics system and flight management system. The Model-based Systems Engineering (MBSE) method is effective to support the design and verification of the aircraft PHM system. As a powerful semantic web construction method, ontology has been widely used to express design information, such as the concepts and the relationships between them. However, traditional graphical MBSE models have a natural weakness in transforming into ontology. In this paper, a semantic MBSE method is proposed to support the transform of the ontology model. Firstly, according to the design characteristics of the aircraft PHM system, a meta-model library of the aircraft PHM system is developed to support the design and evaluation. An MBSE modeling method based on requirement analysis, function analysis, logical architecture design, and physic architecture design is applied in the PHM design process. Secondly, the semantic system modeling language KARMA based on “graph, object, property, point, relationship, role, and extension” (GOPPRRE) is used to transfer the graphical MBSE model to the semantic MBSE model, which can be easily transformed to an ontology model. Finally, an ontology based on semantic modeling is developed to describe the MBSE entities and to support MBSE design. In this paper, a case study of an aircraft fuel PHM system is carried out to validate the proposed method. Based on the developed meta-model library, a complete MBSE design process for the aircraft PHM system is realized. And then an ontology model supporting PHM system design is generated from the semantic MBSE model. The MBSE ontology provides a shareable capability to help designers communicate effectively. Quantitative analysis based on ontology is also provided to verify the complexity and scale of the MBSE design process. Moreover, logical reasoning ability can also be provided to support the early requirement traceability for MBSE design. In general, the case study results show the feasibility and effectiveness of the proposed method for the aircraft PHM system design.

Keywords: aircraft PHM system, model-based systems engineering, KARMA, semantic modeling, architecture design

1 INTRODUCTION

The Prognostic and Health Management (PHM) systems of an aircraft play an important role that can improve the mission completion rate and equipment reliability. The current aircraft PHM system has been developed to the fourth-generation stage. The advancement and complexity of equipment have been greatly improved and integrated, and the fault tolerance is stronger. To provide such a capability, the PHM system should accurately monitor the operational and reliable conditions according to the data from many different systems (Xu et al. (2014)). As a complex system, the PHM system is coupled with other systems, such as the avionics system, flight management computer system, and oil supply system. As a result, the system engineering approach is an effective way to ensure a complete and efficient design of an aircraft PHM system (Dumargue et al. (2016)). However, with the complexity of PHM systems increasing, traditional document-based systems engineering has made it difficult to meet the design requirements and to ensure data consistency during the lifecycle. As an advanced method, Model-based Systems Engineering (MBSE) has become a promising method to guide the PHM system design and to address the challenges brought by its increasing complexity (Hu et al. (2022); Ye et al., 2020). However, different stakeholders have different viewpoints. They design the PHM system by different modeling languages during the MBSE design process (Wang et al. (2019)). The integration of these heterogeneous modeling languages is a huge challenge for implementing MBSE.

Besides, the design process of the PHM system contains lots of structured and unstructured information. Semantic technology is a rule-based data representation method that can manage and reuse information during the PHM design process. Ontology, as a semantic formalize technology, can provide the ability to analyze and evaluate the design information (Meersman (1999)). However, previous MBSE models based on graphical specifications have a huge challenge in transforming into an ontology model. Thus, an ontology development approach based on semantic content is proposed to realize the integration of the MBSE design information (Lu et al. (2021)).

In this work, an ontology-based approach is proposed to support the MBSE design for an aircraft PHM system. A semantic modeling method is proposed to support the MBSE design by converting MBSE models into ontology models. A semantic modeling language, KARMA, is used to construct a domain-specific meta-model library for PHM system design. Based on the meta-model library, a modeling framework including requirement analysis, function analysis, logical design, and physical architecture design (RFLP) is applied to realize the MBSE design of a PHM system. Moreover, based on the semantic modeling, an ontology containing all design information can be generated and used as a sharing vocabulary for the PHM system. With the query and reasoning capability of ontology, quantitative analysis and logical reasoning can be carried out to help engineers develop the PHM system. Finally, a case study about the PHM system of an aircraft fuel system is carried out. In the case study, a design process for the PHM system is realized based on the proposed semantic modeling method, which includes the whole stages of the RFLP framework. Then the ontology model is generated from the MBSE models based on the semantic

modeling method. With the query results of ontology in this case, a quantitative analysis of the PHM system is carried out, which can help designers to determine the scale of the PHM system. A logical reasoning test is also conducted to verify the requirement traceability. According to the results of the case study, the validity and feasibility of the proposed method were verified.

2 LITERATURE REVIEW

2.1 Prognostic and Health Management System

PHM is a brand-new health management solution using the latest research results in modern information technology and artificial intelligence technology. It has very broad applications in both military and civilian fields (Kalgren et al. (2006)). In the aircraft field, the working conditions are changeable, and transition states occur frequently. The difficulties in fault diagnosis are embodied in: 1) The equipment structure is complex and the system integration is high. 2) The service environment is harsh and the working status is changeable. 3) The type of data is rich and scattered. As a multi-domain and comprehensive technology, the PHM system should not only have fault diagnosis capabilities but also make full use of condition monitoring data and fault model information. For this reason, various fault models and intelligent algorithms have been introduced to realize several functions, such as equipment condition monitoring, fault diagnosis, fault prediction, and maintenance decision-marking. From the macro to the details, the levels of PHM system design can be divided into: 1) Design integration and verification of a PHM System. 2) Diagnosis and prediction of subsystems. 3) Simulation and experiment based on data acquisition and the design of test software. The existing PHM system design is mainly concentrated on the third layer, which includes the modeling of the PHM system and the testing of the PHM airborne software. In the last few decades, the diagnostic technology of the PHM system has been widely investigated, especially in the diagnosis model development aspect. In general, the fault characteristics are strongly correlated with the parameters in the mathematical model of an aircraft. Therefore, when conducting the relevant research on the aircraft PHM system, many scholars have carried out in-depth detailed research on the characteristics of the fault evolution process. The diagnosis mechanism model is continuously improved to achieve higher prediction accuracy. The advantage of this method lies in the in-depth exploration of the fault mechanism, and it has good interpretability and credibility in diagnosis and prediction results. Among them, the Kalman filter, particle filter, and other algorithms are widely used in the research of PHM systems based on mechanism characteristics (Pourbabae et al. (2016); Dumargue et al., 2016; Yang et al., 2015). However, because the composition of the aircraft system is too complex and some mechanism characteristics are not clear, it is difficult or even impossible to establish an accurate mechanism model for a PHM system. There are many sensors inside the aircraft system. Fault information can be mined from this data. The method of information mining and prediction based on a large amount of sensor data is called data-driven technology. The difficulties of complex systems modeling can be solved by mining the hidden information from

TABLE 1 | Comparison of PHM technologies.

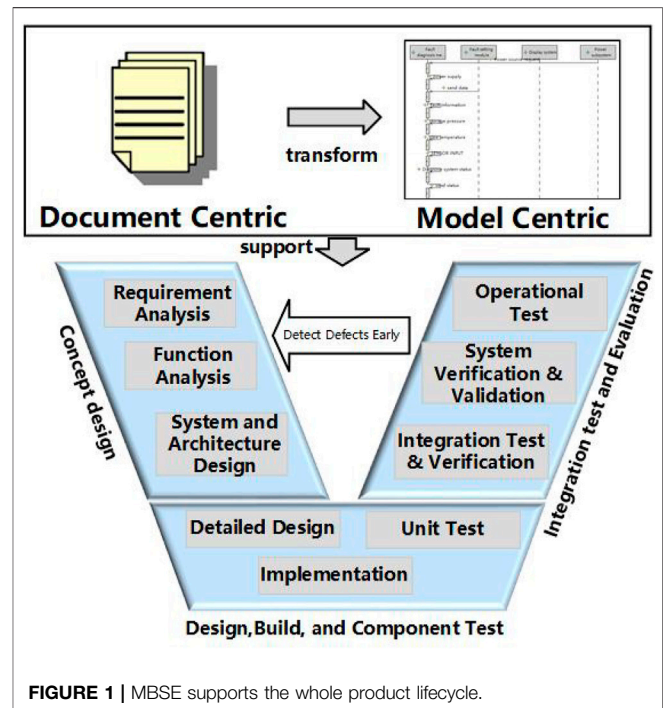
Types	Description	Principal methods	Evaluation
Mechanism	Establishing model based on the internal mechanism	1.Kalman filter (Pourbabaei et al. (2016); Dumargue et al. (2016); Kulkarni et al. (2018)) 2.Particle filter (Yang et al. (2015))	It has good credibility and interpretability. It cannot guarantee the accuracy due to some unclear mechanism and simulation speed is slow due to complex equations
Data-driven	Training the model from a large amount of monitoring data and fault samples	1.Artificial neural network (Zhao et al. (2017)) 2.Fuzzy mathematics theory (Zaidan et al. (2015)) 3.Support vector machine (Sutrisno et al. (2012))	It is suitable for system with unclear principles. However, it relies on the quality of training data. This method cannot be used in the absence of data to support it
Hybrid	Combining the advantages of both mechanism model and data-driven model	1.Data-driven is added to modify the mechanism model. (Song et al. (2017)) 2.Data-driven and mechanism models are paratactic and fused. (Liao and Köttig (2016))	It can guarantee both accuracy and speed. it is still necessary to understand the internal mechanism information

historical data. At present, the algorithms used in data-driven modeling mainly include neural networks, bayesian networks, and support vector machines (Sutrisno et al. (2012); Zaidan et al., 2015; Zhao et al., 2017). However, it is also difficult to accumulate a large amount of data at the initial stage of working. Some researchers have combined the advantages of the above two technologies and proposed a hybrid modeling technology (Song et al. (2017); Che et al., 2018). This method directly establishes a mapping relationship between the system model and the health status based on the measurement data, which is used to correct the mechanism model. Through the hybrid modeling method, most of the mechanism models are retained while introducing actual data for correction, which can ensure both accuracy and speed. The comparison of the three PHM technologies is shown in **Table 1**.

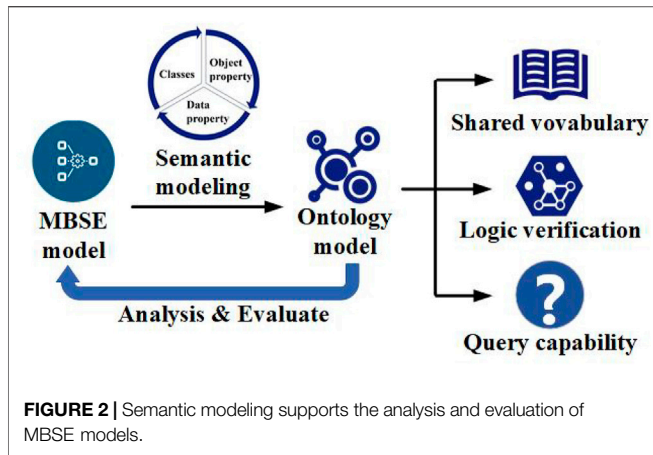
Although many efforts have been made to develop the performance analysis models for fault diagnosis and prognostics during the PHM system design, however, the PHM system is a much more complex system in terms of multi-domain, highly coupled, and multi-function. The PHM system design process should be step-by-step implemented, beginning with requirement analysis, rather than only developing the performance analysis models. The design of the PHM system should pay attention to the following aspects: 1) Multi-level demand analysis and decomposition, multi-view task decomposition and delivery. 2) Standardized full-dimensional complex system model management. 3) Multi-domain coupling and management during the whole lifecycle. As a result, a system engineering approach is required to support the effective design of the PHM system.

2.2 Model-Based Systems Engineering

The complexity of modern aircraft is getting higher and higher. The goals or tasks of an aircraft are also increasing day by day. These have led to the increasing complexity of related hardware and software design. The PHM design process involves a lot of software in different domains. Therefore, a unified modeling standard has become a major issue in the aircraft PHM system design. Traditional document-based systems engineering is widely used in complex system design. However, it is difficult to meet the verification and validation requirements when the products become more and more complex. With the continuous expansion of information, traditional document-based systems engineering has the

**FIGURE 1** | MBSE supports the whole product lifecycle.

disadvantages of inconsistent information, incorrect versions, and poor readability during the product iteration process. In order to solve the shortcomings of traditional document-based systems engineering, the International Council on Systems Engineering (INCOSE) proposes the Model-Based Systems Engineering (MBSE) method, as shown in **Figure 1**, and gives its definition (Friedenthal et al. (2007)): “MBSE is a formal application of a modeling method, which supports system requirements, design, analysis, verification, and validation activities.” MBSE aims to transform systems engineering from document-centric to model-centric. It is a guide for designers to conduct activities during the design process. These activities start from the conceptual design stage and continue throughout all subsequent lifecycle stages, including build, component testing, integration testing and evaluation. From the current research results, the main design process of MBSE includes requirement



analysis, function analysis, logic design, and physical architecture design (Wang et al. (2016); Bindschadler et al., 2017; Allen, 2016). In the process of system requirements analysis, the requirements are refined according to the functional requirements, design requirements, and expected goals. During the process of functional analysis, activities are proposed to meet the requirements of each item. The logical architecture model is constructed by the information flows among these functional activities. The physical architecture model is a complete description of all the components of the complex system. Based on the process of RFLP, different engineers can design and optimize the model at the same time based on the same system architecture. Data interoperability across the different stages of RFLP is also important for system design. A design structure matrix (DSM) is a mathematical expression method to describe the interaction of design information across the stages. Besides, Model Driven Interoperability System Engineering

(MDISE) is also proposed to manage the interaction relationship by developing conceptual models (Zacharewicz et al. (2020).

With the continuous refinement and the further deepening during the system development process, the MBSE is regarded as the future of system engineering methods. Many fields are constantly proposing different implementation methods for the application of MBSE. MBSE provides a modeling specification for a standardized workflow during the design process, and the corresponding logic also provides a reference for the implementation and verification of products. At present, there are several mature modeling specifications (Friedenthal et al. (2014); Estefan, 2007; Roques, 2017; Kelly and Tolvanen, 2008): Harmony-SE, Object-Oriented System Engineering Method (OOSEM), and Arcadia. During the product design and development process, MBSE realizes the model expression through a graphical modeling language. This kind of graphical modeling language does not require professional programming language knowledge and can express the design information clearly. The main MBSE modeling languages include UML, SysML, DSML, BPMN, UPDM, and AADL (Torre et al. (2018); Chinosi and Trombetta, 2012; Hause et al., 2012; Feiler et al., 2006). However, the above-mentioned MBSE modeling method is mainly oriented towards professionals. It does not consider collaborations among different fields, nor establish requirement traceability during the design process. When the requirements change, the MBSE design model should be re-developed. As a result, it cannot really improve the efficiency of model development through the collaboration of different fields.

2.3 Semantic Modeling

The design process of a complex system contains a large amount of structured and unstructured information. Structured

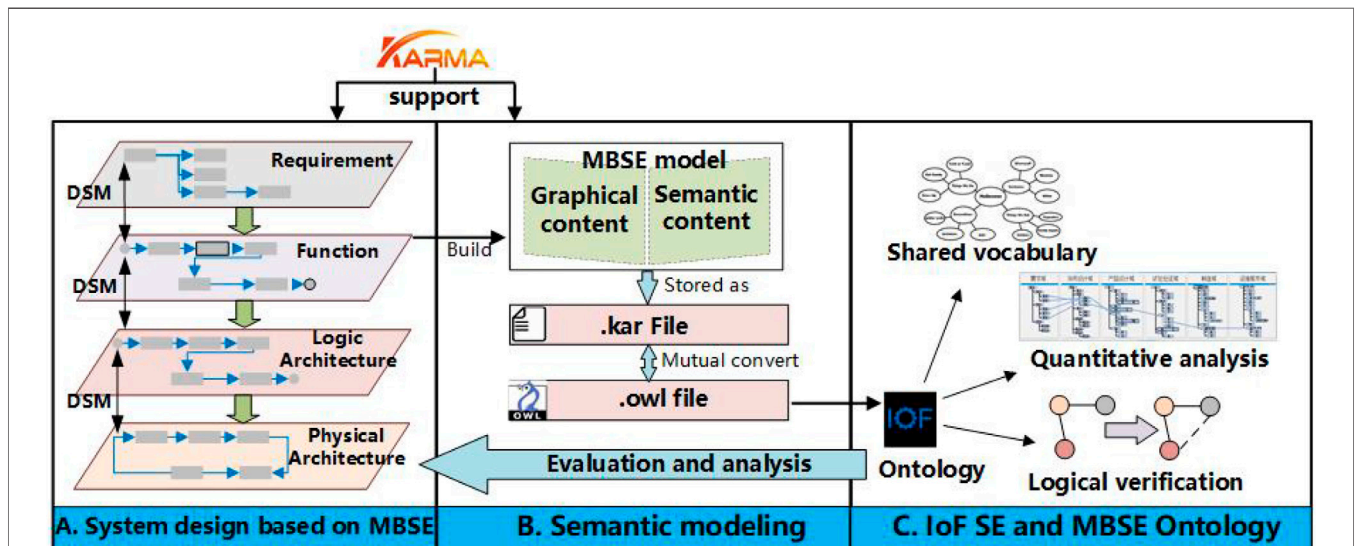


FIGURE 3 | Semantic modeling framework based on KARMA for the PHM system, (A) System design based on MBSE, (B) Semantic modeling, (C) IoF SE and MBSE Ontology.

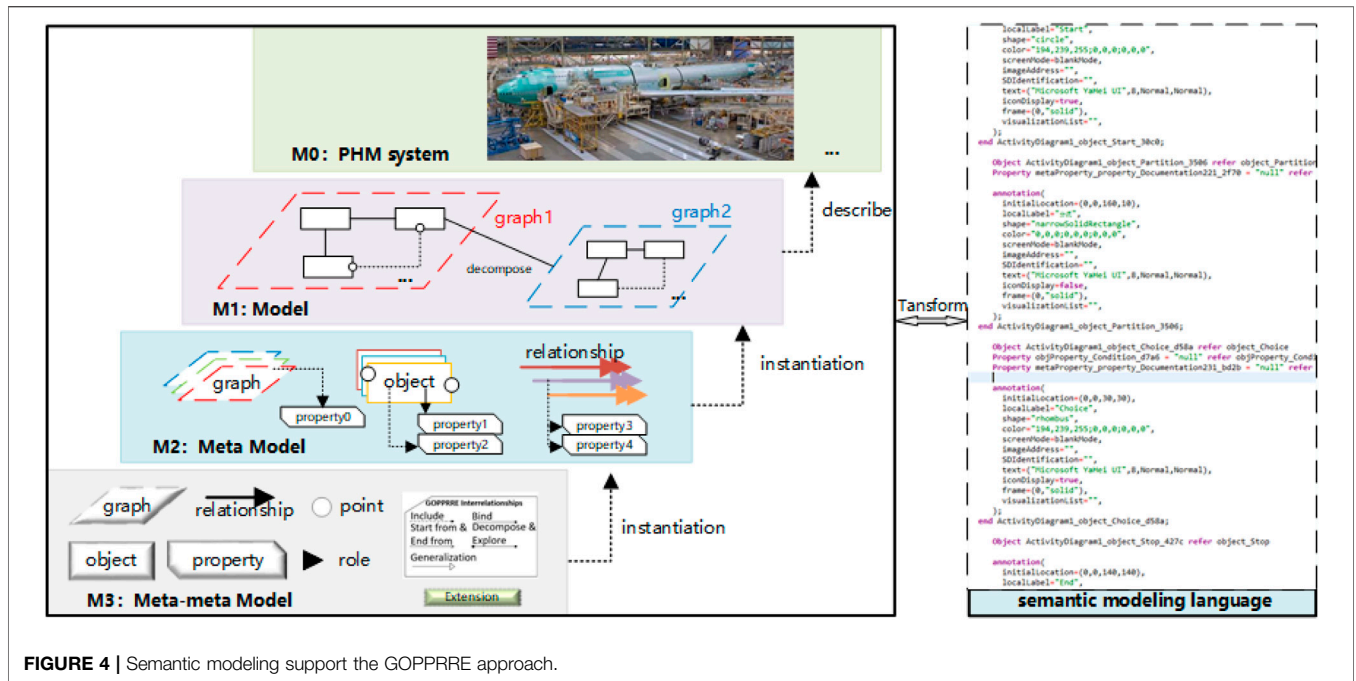


FIGURE 4 | Semantic modeling support the GOPPRRE approach.

information belongs to the data that can be digitized, so it can be used flexibly under the management of computer and database technology. Unstructured information cannot be completely digitized, and there is still a large amount of valuable information in the unstructured data, which can support the design and management of complex systems. Therefore, it is necessary to build a consistent representation mechanism for unstructured data. On this basis, a unified platform is developed to effectively process and search various unstructured information types. In order to cope with the challenges of unstructured information transmission and management, the semantic Web is proposed to transform the Internet into an ordered knowledge base, which can be understood by computers. As a result, the existing information islands can be integrated into a huge database for better management (Ha et al., 2014; Hahm et al., 2015). Semantic technology is a rule-based data representation method that is used to describe data and entities in the real world so that machines can better understand them. In order to realize the function of the semantic web, it is necessary to provide a structured semantic description mechanism that the computer can understand and to develop a series of rules for automatic reasoning. The challenge of the semantic web is to provide a language that can express the data and the reasoning rules. In addition, the language needs to be able to apply the rules to the Semantic Web.

Semantic information modeling is of great significance for managing and reusing information during the design process. Some extended information can also be obtained by reasoning over the semantic information, which is conducive to communication between multiple disciplines or organizations. The model based on Web semantics mainly includes Extensible Markup Language (XML), Resource Description Framework (RDF), and Web Ontology Language (OWL) (Bikakis et al. (2013); Cheng

et al., 2016; McGuinness and Van Harmelen, 2004). The XML data model has the advantages of data storage format, high scalability, self-description, and interoperability. The disadvantage is the limited ability to express complex knowledge. Therefore, XML is only used at a grammatical level to describe the structure and content of data. RDF is a solution to the problem of XML models. RDF provides a formal framework for semantic information. It is more powerful than XML in knowledge representation. The disadvantage is that the semantic description of complex knowledge is not accurate enough, and it has limitations in the representation of complex knowledge.

In order to better describe the information on the Semantic Web, a more powerful ontology modeling language is needed. As a philosophical concept, ontology is an abstraction of the objective world. Recently, as a part of ontology, temporal ontologies (Deng (2018)) have been proposed as an expansion of ontology. At present, ontology has been widely used in the field of computer science to describe concepts and the relationships between them. An ontology is a set of precise descriptive statements about the field of interest. It can effectively prevent misunderstandings in communication, and ensure the software is uniform and predictable. Ontology language is used to display the formal description of the ontology. In (Zheng et al. (2021)), semantic technology is considered as the core of the cognitive digital twins concept because of its capability in data interoperability. By providing additional lexical and formal semantics, OWL provides a higher level of machine interpretability than XML and RDF. OWL describes the semantics of knowledge resources and the connections between them. Compared with XML and RDF, OWL has rich semantic expression capabilities for knowledge sharing. The basic elements of OWL are class, individual, and property. A class is a type of object or kind of thing. An individual represents the instance of a class. A property is a binary relationship between

TABLE 2 | Domain-specific meta-model library for PHM.

Modeling Process	Meta-graph	Meta-object	Meta-relationship
Requirement analysis	Requirement diagram	Package, test case, comments, requirements, constraint modules	Include, trace, reuse, verify, improve, satisfy
	Use case diagram	Package, participant, module, use case, interface module	Include, generalize, associate, Extend
Function analysis	Package diagram	Package, module, feature, interface module, model, model base, value type, constraint module	Generalization, association, introduction, combination
	Activity diagram	Packet, action, shunt, send action, fork, node, select, accept event, start, terminate, end	Object flow, control flow
Logical design	Sequence diagram	Modules, operator, lifeline object, selection, cycle	Deliver message, reply message, create message
	State machine diagram	Fork, start, end, selection, status, historical mark	Conversion
	Fault tree	Bottom event, intermedia event, top event	Connection
	Petri net	Place, transition	Flow relationship
	Markov chain	State	Transition
Physical design	Block definition diagram	Package, interface block, block, constraint block, value type, comment	Generalization, association, project flow, composition
	Internal block diagram	Interface block, Comment, Internal block	Object stream
	Parameter diagram	Parameter, interface block, constraint block	Parameter flow
	PHM composition diagram	PHM block	Connection
	PHM information transmission diagram	PHM equipment	Data flow

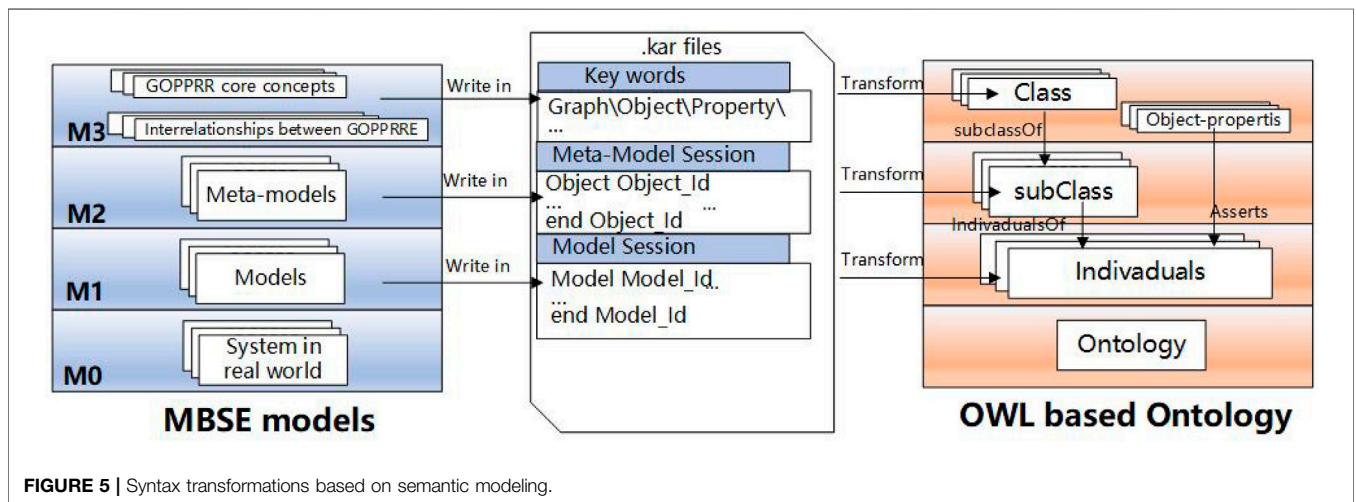


FIGURE 5 | Syntax transformations based on semantic modeling.

individuals, which is used to connect individuals together. The OWL model can effectively describe the transfer relationship, symmetry, uniqueness, and opposite relationships between objects. As shown in **Figure 2**, the ontology model can inherit the logical relationship from the MBSE model and integrate the knowledge graph by semantic modeling method. The ontology model can support MBSE design by providing the shared vocabulary, query capability, and logic verification.

3 SEMANTIC MODEL-BASED SYSTEMS ENGINEERING FOR PHM DEVELOPMENT

3.1 Semantic Modeling Framework for PHM Development

The semantic modeling framework based on KARMA and Industrial Ontologies Foundry (IoF) for the PHM system is shown in **Figure 3**. The whole architecture consists of three

TABLE 3 | Design information in ontology.

Class	Subclass	Individual Example
Graph	The design diagram of PHM system	Use case diagram of maintenance function scenario, Activity diagram for valve fault diagnosis
Object	The basic components in design diagram of PHM system	Design engineer, valve, tank, pilot
Point	The port of component in PHM system	Signal input point, signal output point
Property	The attribute of PHM system	Object condition, object name, object ID
Relationship	The connection type of each PHM design diagram	Control flow, association, connect, fuel flow
Role	The direction of data flow	Input, output, start, end

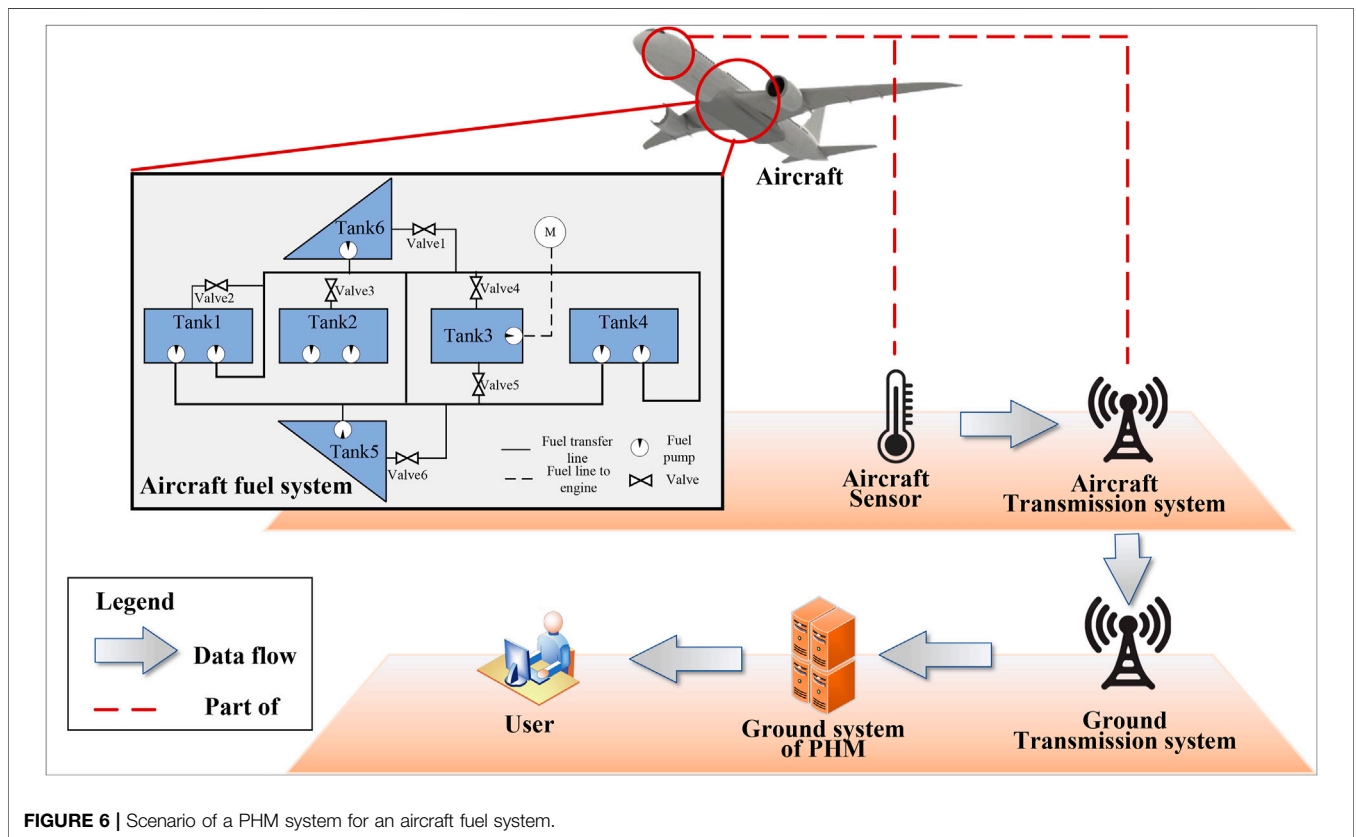


FIGURE 6 | Scenario of a PHM system for an aircraft fuel system.

parts: system design based on MBSE semantic modeling; and IoF SE and MBSE ontology. The system was designed based on the MBSE methodology. In this work, the semantic modeling language KARMA is used to construct the MBSE models during the system design. Then the MBSE models are converted to ontologies by KARMA, which supports MBSE semantic modeling.

During the MBSE design, the OOSEM is used to design the PHM system. The OOSEM method divides the design process into requirements analysis, functional analysis, logical architecture design, and physical architecture design stages. The requirement analysis stage defines what users need. The function analysis stage defines the product features to meet the users' requirements. The logical design stage designs the system's logical components and logical architecture. The physical architecture design stage aims to implement the system architecture from the physical perspective. The

relationship of the across-stage elements can be described by DSM during the design process.

Semantic modeling provides a text-based description for MBSE design. The system designer constructs MSBE models graphically. During the development of the graphical MBSE model, the semantic content is also stored in a text file. In other words, graphical content and semantic content are two of the description methods of the MBSE model. The text containing all the design information is stored in a .kar file. By reading the .kar file, the design information can be extracted and used in the construction of the ontology.

IoF SE and MBSE ontology is used to evaluate and analyze the MBSE models. It was developed by the IoF System Engineering Working Group. As an ad-hoc organization, the mission of IoF is to create a set of core and open reference ontologies that span the entire domain of digital manufacturing. Such ontologies can be created by standard organizations or

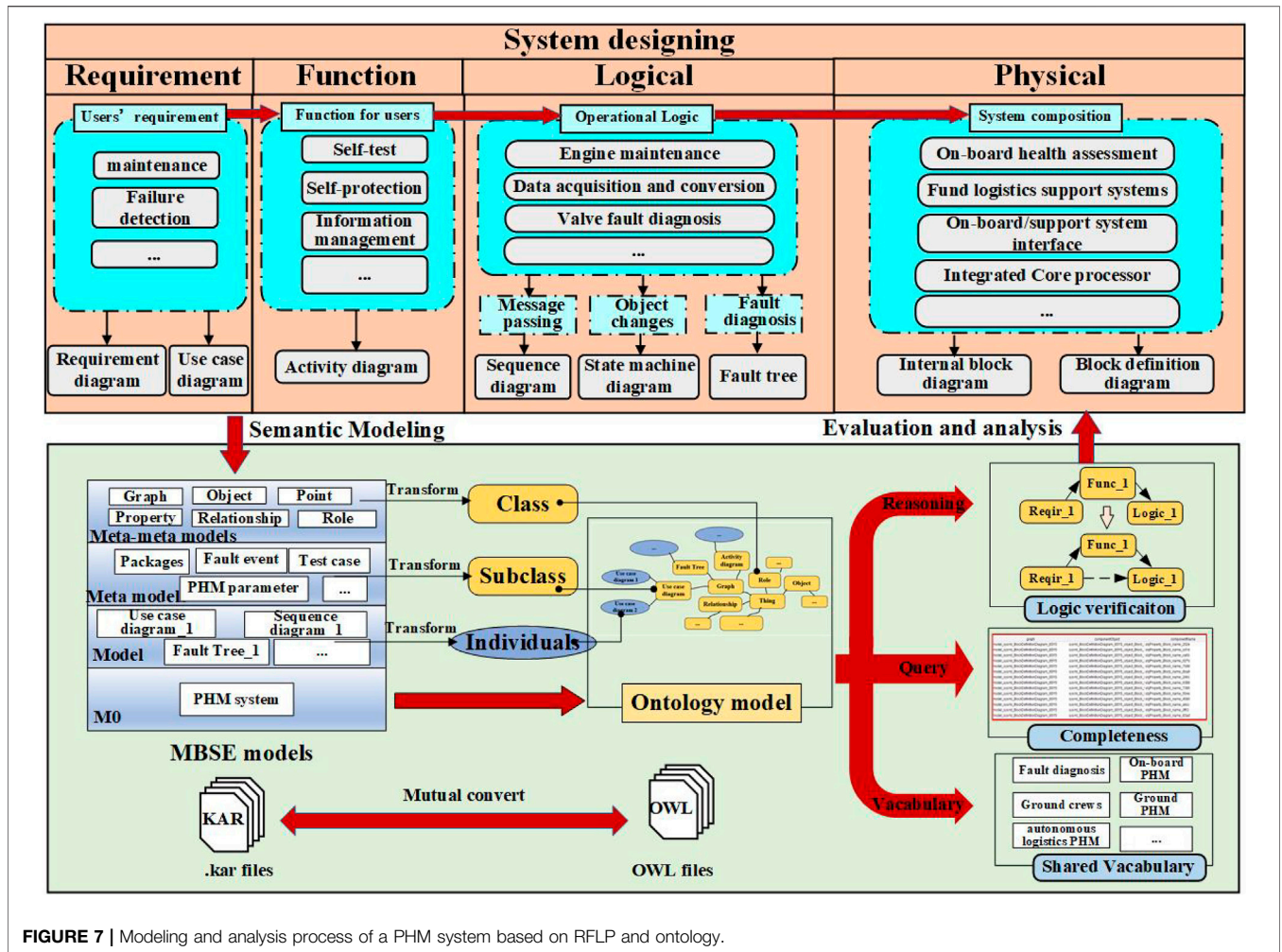


FIGURE 7 | Modeling and analysis process of a PHM system based on RFLP and ontology.

research groups. The suite of open reference ontologies represents a shared vocabulary that can be used across industrial organizations. That means the IoF SE and MBSE ontology can improve communication efficiency during the MBSE design. Quantitative analysis is also an excellent capability for ontology. The ontology can analyze and evaluate the MBSE design model, such as component count, to help designers understand the system better. Besides, automated reasoning can be run over a defined ontology to carry out logical verification. In other words, ontology can make information processing more intelligent and provide the capability of evaluation and analysis for MBSE design.

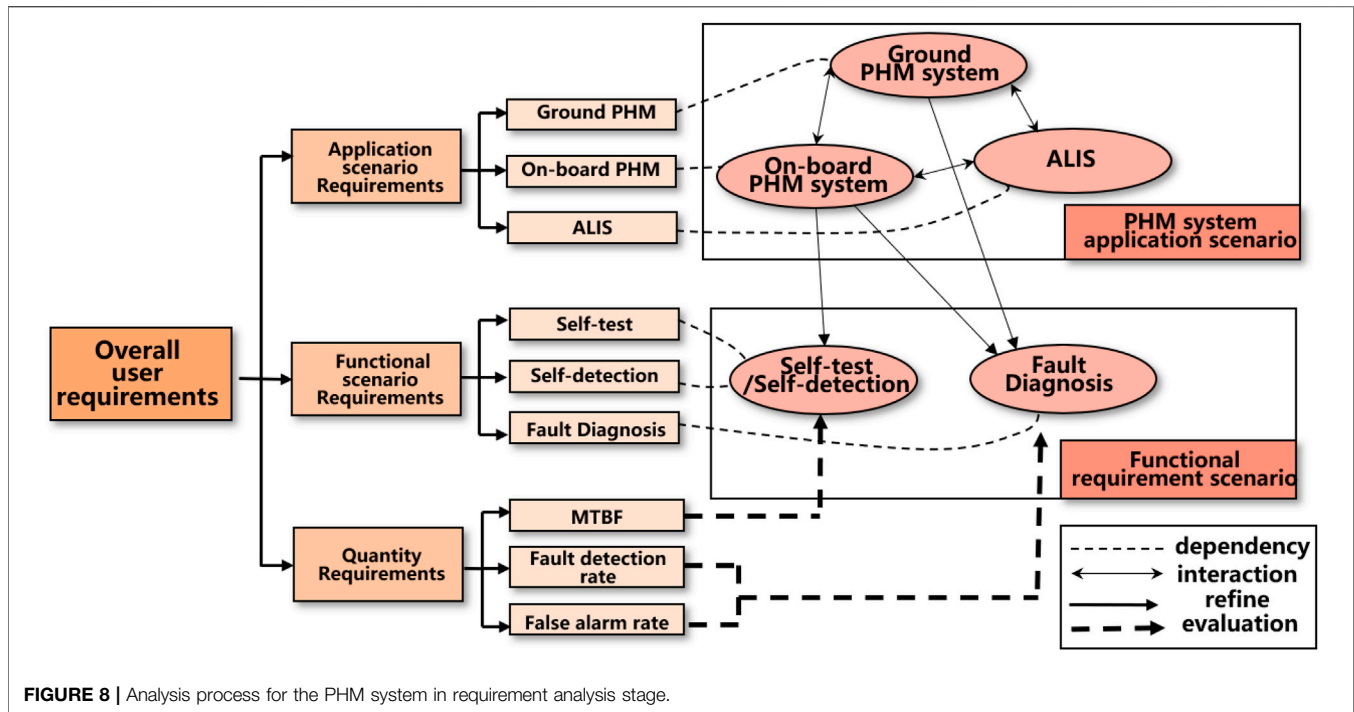
3.2 KARMA Language for PHM

In order to realize the simultaneous construction of multi-domain models, some methods are proposed by extracting the commonalities of different modeling languages. The GOPPRR meta-meta modeling method is one of them. This method provides a more comprehensive representation of the data than other meta-modeling methods. The multi-architecture modeling language KARMA is a semantic modeling language based on the GOPPRR meta-meta

modeling method. Related technical research on KARMA began in 2015. The KARMA research team conducted a systems engineering practice, making the proposed GOPPRR methodology more suitable for solving system engineering problems.

The proposed semantic modeling approach is shown in Figure 4. A Meta Object Facility (MOF) modeling framework (M0-M3) proposed by the Object Management Group (OMG) is applied to develop the MBSE models. The description of each layer is shown as follows:

- M3: Meta-meta models refer to the basic elements of the constructed model compositions and their interconnections. In this study, GOPPRR meta-meta models are applied to support the construction of meta-models. Besides, the extensions of the GOPPRR meta-meta models are also introduced.
- M2: Meta-models include the basic model compositions and their connections. Models can be developed by using meta-models.
- M1: MBSE models are developed to represent real-world systems based on meta-models.
- M0: Real-world artifacts PHM systems are considered.



The GOPPRRE approach is extended by the GOPPRR approach to construct the MBSE model syntax and semantics. In Lu et al. (2021), a new constraint concept is defined as the connecting patterns within abstract syntax. The details of the GOPPRRE concepts in **Figure 4** are introduced:

- A graph refers to a collection of objects, relationships, and roles. The graph can be an independent visual diagram. A graph can be not only a stand-alone visualization diagram but also a virtual graph that represents the relationships of other objects by decomposition and explosion.
- An object is the basic block element of a graph, for example, an action block in a SysML activity diagram.
- An object point refers to the port of an object, which represents the attachable connector in the object.
- A property is an attribute description of the other five meta-meta models.
- A relationship is one connection between the points and/or objects, such as the control flow in an activity diagram.
- Role is used to define the connection direction of relationship. A relationship is associated with two roles. Through the role, the relationship is defined as one that is bound to one point or one object at its end.
- Extension refers to the additional constraints used to construct meta-models. It refers to one connected relationship between one point or object and one role at the end of the relationship.

During the PHM system development, the domain-specific meta-model library can be generated to support the RFLP modeling process by KARMA as shown in **Table 2**. The meta-graph is the required

models constructed during the PHM system design process. A meta-object is the component of a meta-graph. Meta-relationship is the basic connection type of meta-graph. The requirement model can be constructed by the requirement diagram and use case diagram. The use case diagram describes the stakeholders and use cases of a PHM system. The package diagram and activity diagram can be used to support the function analysis. The package diagram is used to describe the hierarchy of PHM system components. The activity diagram describes the execution process of system behavior. The logical model consists of a state machine diagram, a sequence diagram, a fault tree, a Petri net, and a Markov chain. The state machine diagram can express the system state and the state transformation. The sequence diagram describes the time interaction of the activities in the system. The Petri net, Markov chain, and fault tree can be used as the fault diagnosis methods in a PHM system. The physical model can be expressed by a block definition diagram, internal block diagram, parameter diagram, PHM composition diagram, and PHM information transmission diagram to describe the physical implementation structure of the product.

3.3 Semantic Models for PHM

The semantic modeling method constructs the MBSE model based on semantic specifications instead of graphic specifications. As shown in **Figure 5**, *.kar* files are developed as the intermediate files between MBSE models and ontology models. The GOPPRR concepts have been defined as the key words in the KARMA language. All the meta models in M2 and models in M1 can be written in different *.kar* files. These *.kar* files can be transformed to OWL based on ontology models. The details of the KARMA language syntax specification are introduced in Gray and Rumpe (2021), which is shown as **Algorithm 1**. The Model class can be considered as the collection of object classes, relationship classes, role classes, and property classes. The

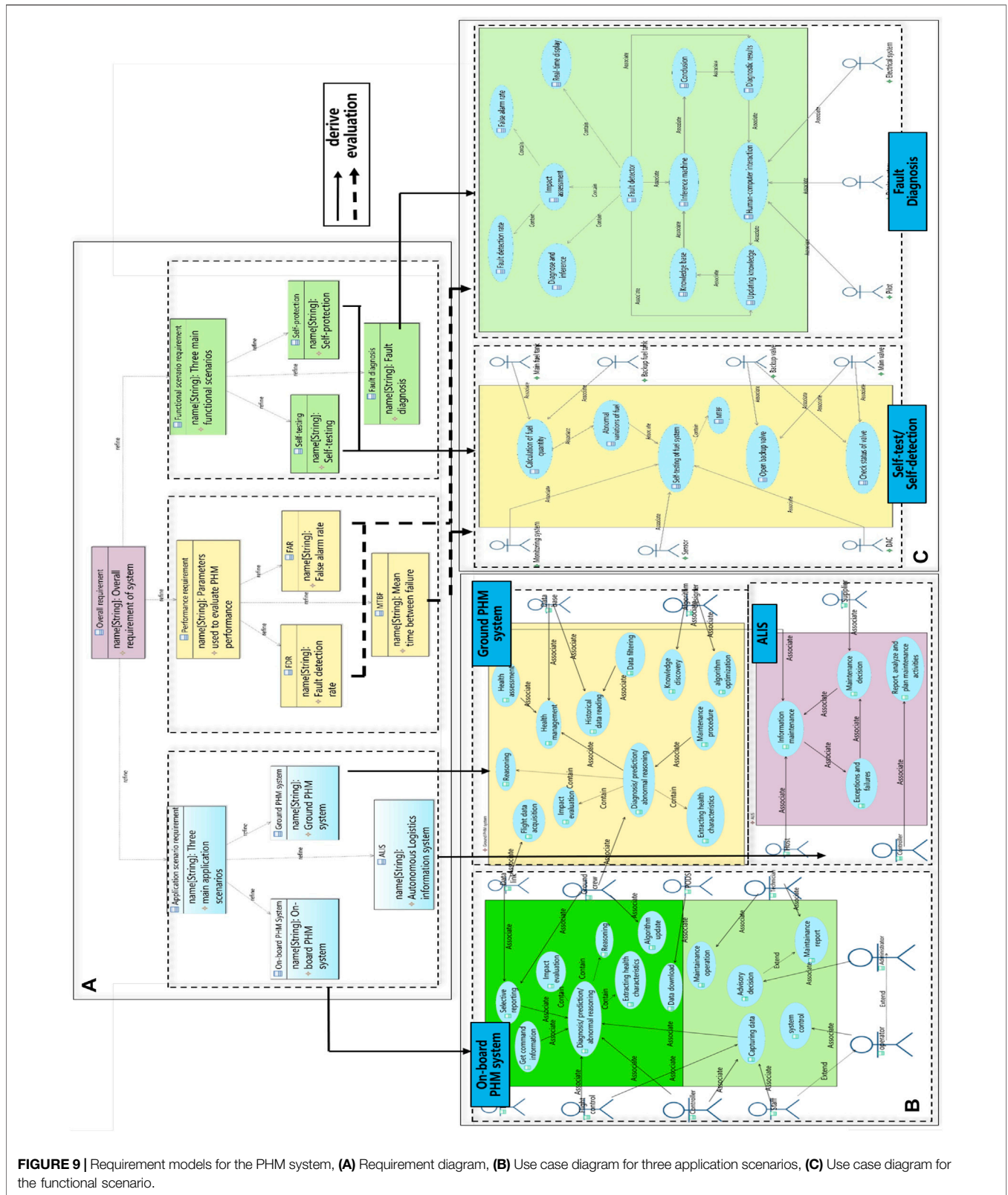


FIGURE 9 | Requirement models for the PHM system, (A) Requirement diagram, (B) Use case diagram for three application scenarios, (C) Use case diagram for the functional scenario.

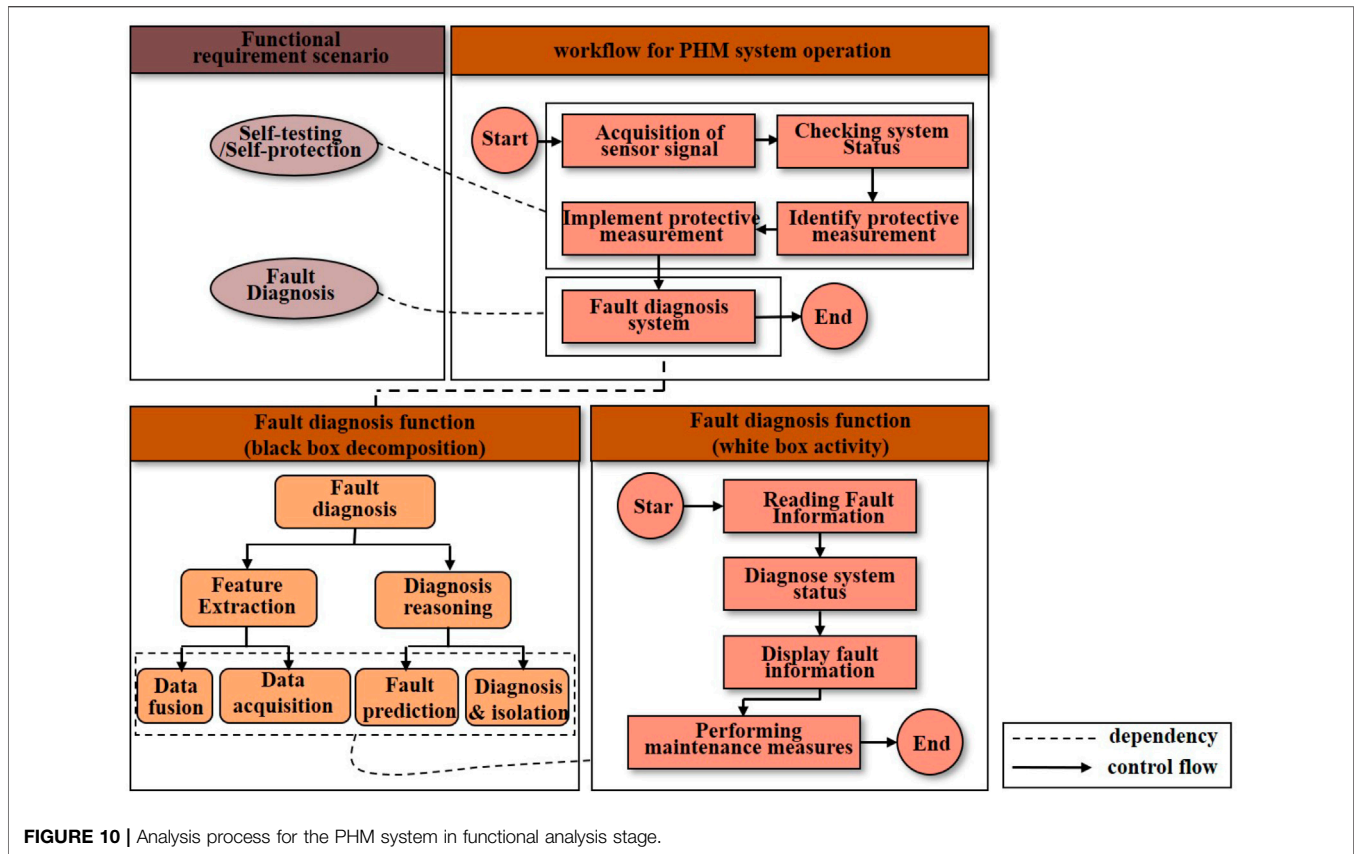


FIGURE 10 | Analysis process for the PHM system in functional analysis stage.

annotation of each class instance contains the basic description information, such as label and location.

Algorithm 1. KARMA semantic model content

```

1: partial Model {Model_ID} instanceof {metaModel_ID}
2: //describe the object in the model
3: Object {Objec_ID} refer {metaObject_ID}
4: //some annotation of the object
5: annotation(
6: ...
7: );
8: end {Objec_ID};
9:
10: //describe the Relationship in the model
11: Relationship {Relationship_ID} refer {metaRelationship_ID}
12: //some annotation of the relationship
13: annotation(
14: ...
15: );
16: end {Relationship_ID};
17:
18: //describe the Role in the model
19: Role {Role_ID} refer {metaRole_ID}
20: //some annotation of the role
21: annotation(
22: ...
23: );
24: end {Role_ID};
25:
26: //describe the Property in the model
27: Property { Property_ID} refer {meta Property_ID}
28: //some annotation of the role
29: annotation(
30: ...
31: );
32: end {Role_ID};
33:
34: //define the connection between Objects or Roles
35: Connect(connector({Object_ID},{Relationship},{Role_ID}),connector({Object_ID},
36: {Relationship},{Role_ID});
37: //the annotation of this model
38: annotation(
39: localLabel="model",
40: );
41: end {Model_ID};

```

Based on the syntax of KARMA, the graphical models can be expressed with semantic content. This semantic content is easy to convert to OWL based on ontology by semantic specification. A workflow for transforming the .kar files to ontologies are also demonstrated in Figure 5. The class for each GOPRRR concept represents the meta-meta models (i.e., Graph, Object, Relationship, Role, Property, and Point). Their interrelationships are transformed to object property concepts in the ontology. Meta-models are then transformed into subclasses in the ontology. Models are transformed into individuals based on their related subclasses. Based on the object property concepts, the interrelationships among individuals are defined. Moreover, the data property is used to define the value of each property. The data property type in ontology is used to define the data type of each attribute. Finally, the MBSE models representing the real world are transformed into the ontology containing individuals, data properties, and object properties. MetaGraph 2.0 is used to develop the MBSE models. In MetaGraph 2.0, the KARMA model and the IoF SE ontology can be mutually converted. The ontology model generated from MetaGraph 2.0 can express the semantic content of the KARMA language as an MBSE model structure. The detailed transform algorithm from KARMA to OWL is shown as Algorithm 2.

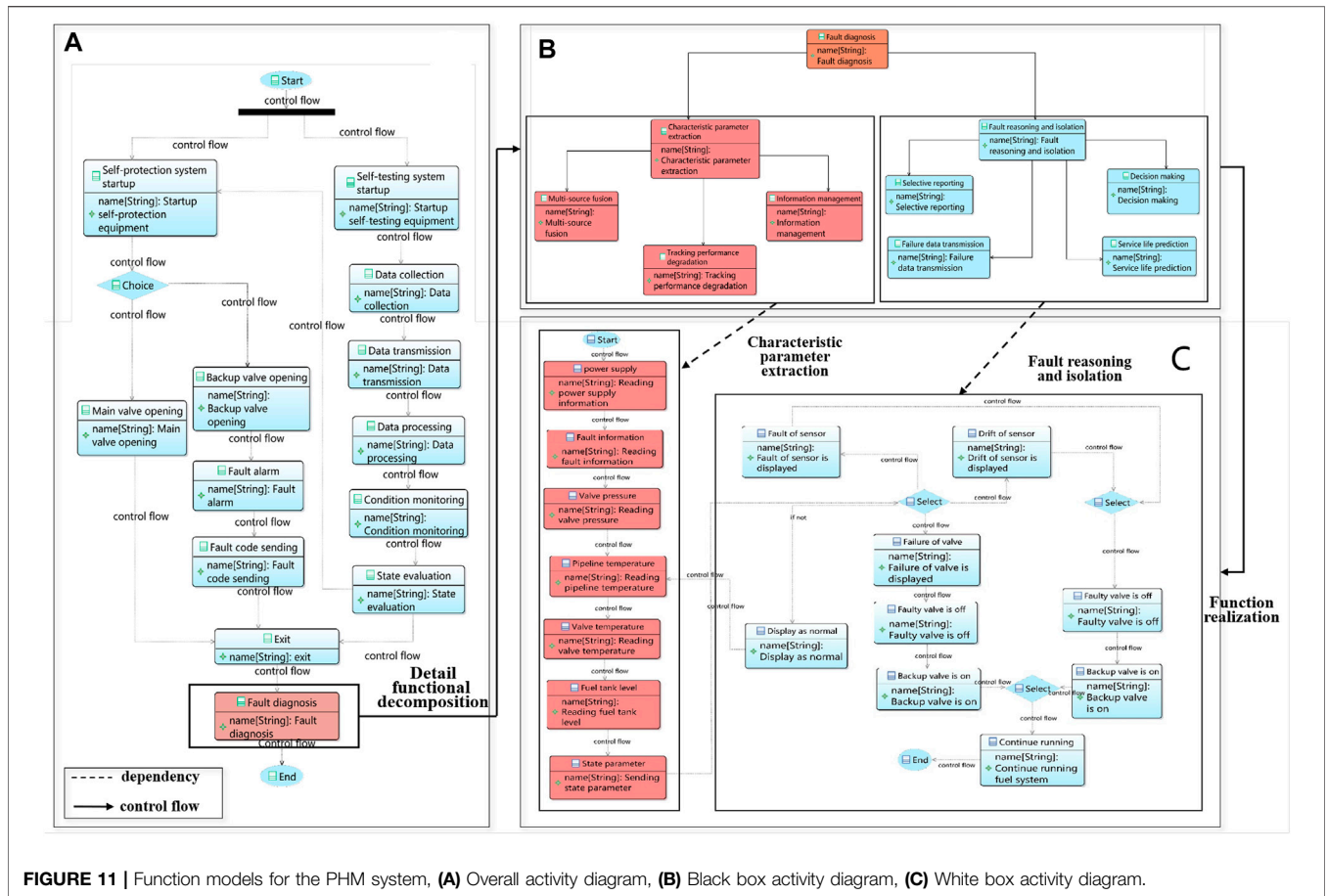


FIGURE 11 | Function models for the PHM system, (A) Overall activity diagram, (B) Black box activity diagram, (C) White box activity diagram.

Algorithm 2. transformation from KARMA to OWL

```

1: function Karma2Owl:
2:   //for each language in karma folder
3:   for(IResource fileLanguage : karmaFolder){
4:     //for each Element meta model in file folder "meta{Element}", where {Element} can be replaced
5:     by Graph/Object/Property/Point/Role/Relationship
6:     for(IResource f : filemeta{Element}){
7:       //read meta{Element}
8:       Meta{Element} meta{Element} =
9:       karmaCompilerManager.parseMeta{Element}ById(f.getName());
10:      ...
11:      //generate {Element} meta model in OWL file
12:      OntClass owlmeta{Element} = ontModel.createClass(meta{Element}.getId()); ...
13:    }

```

The transformed ontology contains all the design information during MBSE design. The various concepts in the PHM system are represented by entities in ontology. The subclass layer defines all the meta models, including the basic components, the basic attributes, and the connection types. Individuals are instantiated entities that contain all the model information of the PHM system. The detailed PHM design information in ontology is shown in Table 3.

4 CASE STUDY

4.1 Problem Statement

A case study is conducted, in order to validate the proposed approach. In this paper, a PHM system for an aircraft fuel system

is designed by the proposed approach. The scenario of the PHM system for an aircraft fuel system is described in Figure 6. The whole system is split into two main parts: the airborne part and the ground part. The relevant parameters of the fuel system are monitored by sensors and transmitted to the ground through the transmission system. The ground processor analyzes the data and interacts with the users. As shown in Figure 6, the aircraft fuel system is mainly composed of fuel tanks, fuel pumps, valves, and fuel pipelines. Fuel tanks are located at the front, rear, and wings of the fuselage. These fuel tanks are connected through fuel pipelines (Zhao et al. (2022)).

4.2 Semantic Models for PHM Systems

On the basis of the domain-specific meta-model library described in Table 2, the RFLP design process is applied to implement the PHM MBSE design. Figure 7 shows the schematic diagram of the semantic ontology method supporting the PHM system’s MBSE design. In the requirement analysis stage, the users’ requirements about the PHM system, such as maintenance and failure detection, are considered by the requirement diagram and use case diagram. In the function analysis stage, an activity diagram is used to define the functions and services to meet the requirements. In the logical design stage, the logical architecture and interfaces of the system are designed to realize the PHM system functions. A sequence diagram, a state

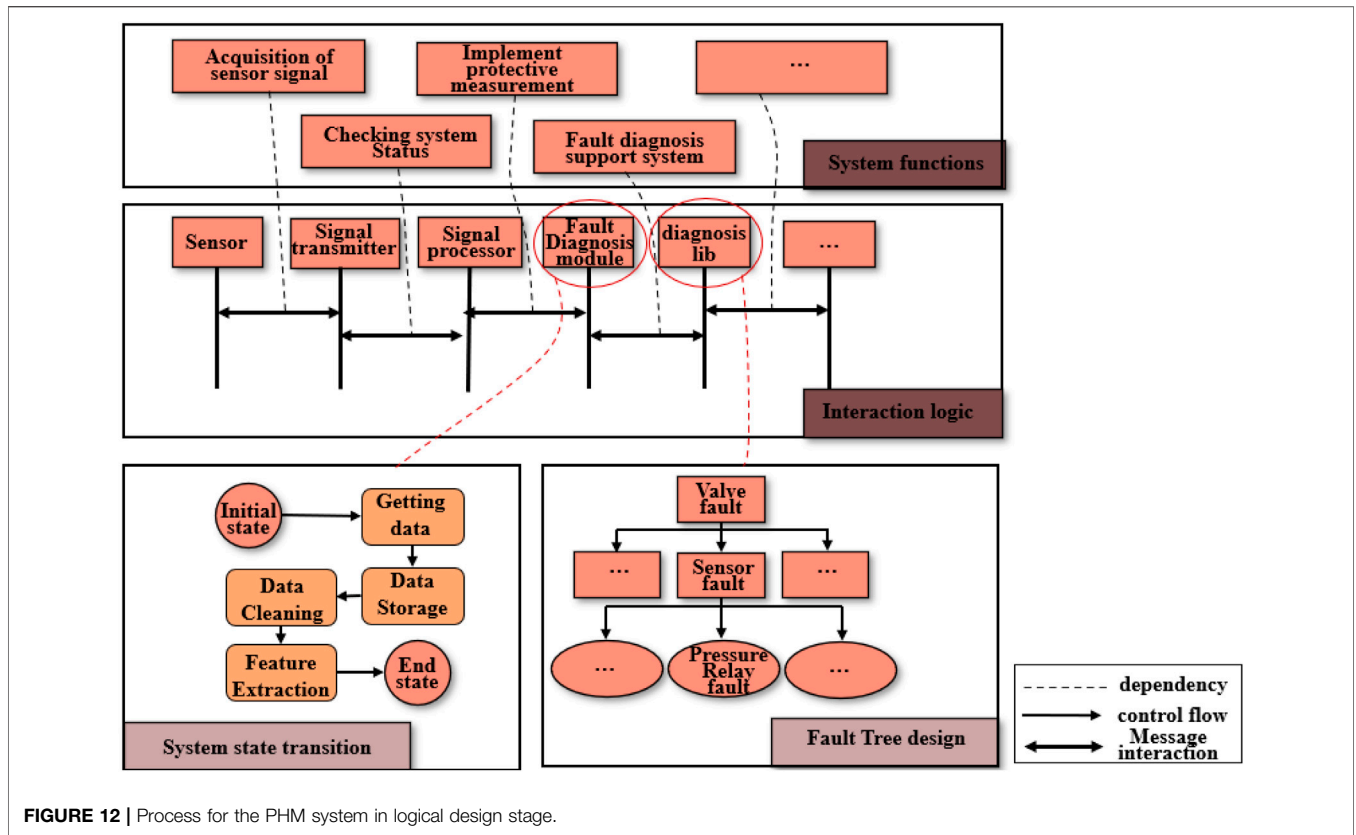


FIGURE 12 | Process for the PHM system in logical design stage.

machine diagram, and a fault tree diagram are developed in this stage. An internal block diagram and a block definition diagram are used to define the physical architecture of the real PHM system. Based on semantic modeling, the graphical MBSE models can be transformed into ontology models. Then the ontology model can be used to evaluate and analyze the system design process according to the reasoning and query ability of the ontology. Besides, ontology can provide a shared vocabulary to improve the communication efficiency during the PHM system design process. The detailed MBSE design process is introduced as follows.

4.2.1 Requirement Models for PHM System

The requirements model is located at the top of the RFLP method and consists of various requirements items. All the requirement items proposed by the stakeholders are displayed through the requirement diagram, as shown in Figure 8. The overall requirements of the PHM system are first refined into three sub-requirements: application requirements, functional requirements, and performance requirements. Then the above sub-requirements are further refined. The application requirements consist of requirements for on-board systems, ground systems, and autonomous logistics information systems. The functional requirements include self-testing, self-protection, and fault diagnosis. The performance requirements are refined into the mean time between failures, fault detection rate, and false alarm rate. Each proposed requirement item has its own corresponding functional scenario or application scenario. The detailed refinement

of the requirements model is shown in Figure 9A. In order to realize these requirements, the scenario use case diagrams need to be built. Use case diagrams in the model are used to show all the actors and use cases in the PHM system. Figure 9B shows the use case diagrams of the three application scenarios. The use case diagram of the on-board PHM system includes the pilot, portable data storage, flight controller, data links, controllers, ground crew, and commands between the above use cases. The use case diagram of the ground PHM system consists of ground crew, data link, data base, and algorithm designer and their commands and operations. The use case diagram of the autonomous logistics information system contains host, controller, and supplier and their commands and operations. Moreover, human behavior is also considered in the on-board PHM system scenario (Zacharewicz et al. (2020)). This use case classifies the related humans into staff, operator, administrator, and technician. The associated use cases include advisory decision, capturing data, system control, and maintenance operation. By summarizing the above four parts, the application scenario use case diagram of the overall PHM system is obtained. In addition to the application scenarios, the operating mechanism of the PHM system is also shown in the functional scenario use case diagram (see Figure 9C). It shows the use cases of self-testing, self-protection, and fault diagnosis. The working components in the functional scenario use case diagram for self-testing and self-protection are sensors, monitoring system, digital to analog conversion system, main valve, main fuel tank, backup valve, and backup fuel tank. The fault detector of the PHM system consists of diagnosis and reasoning, impact assessment, and real-time display. Use cases for a complete

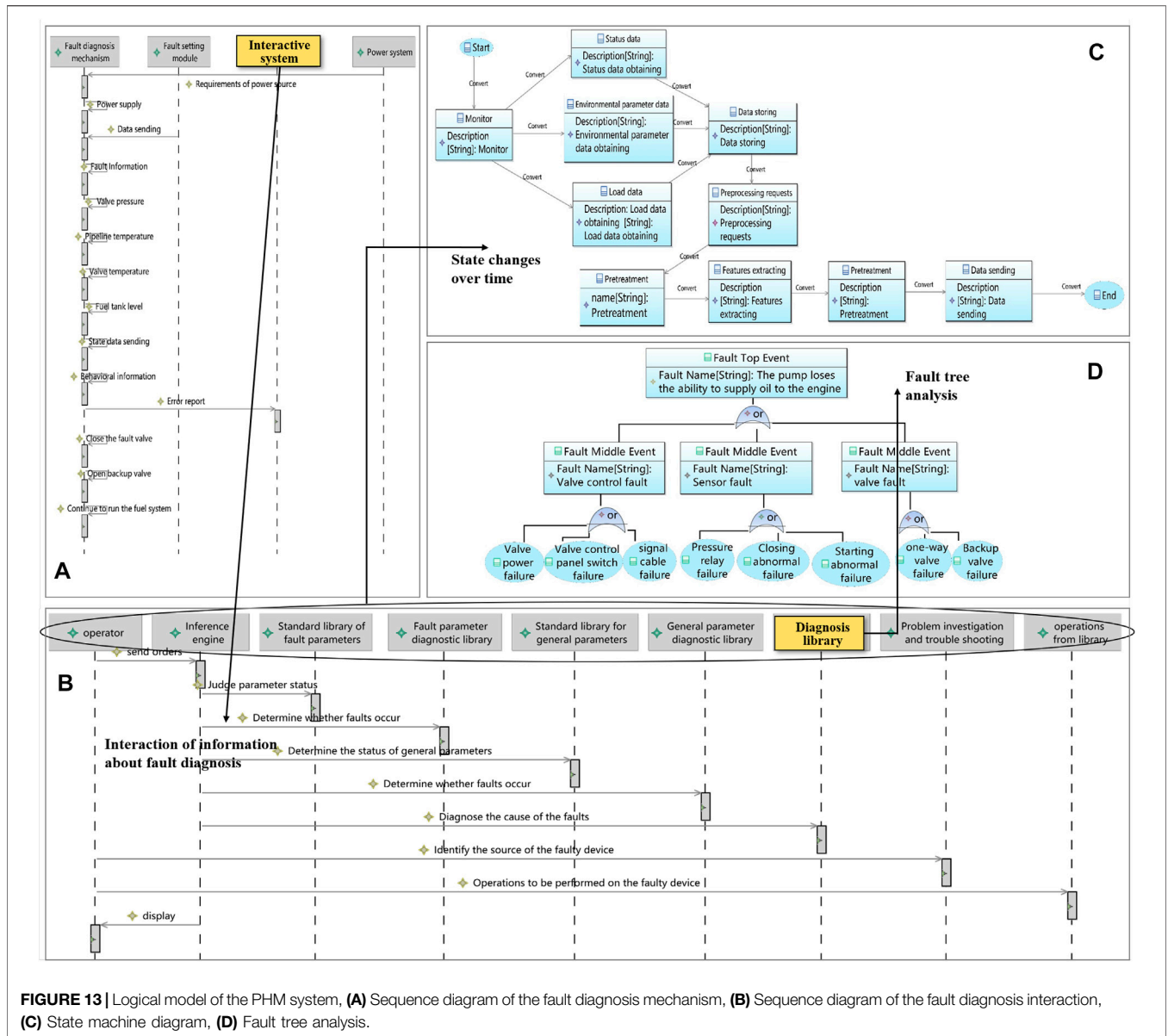


FIGURE 13 | Logical model of the PHM system, (A) Sequence diagram of the fault diagnosis mechanism, (B) Sequence diagram of the fault diagnosis interaction, (C) State machine diagram, (D) Fault tree analysis.

fault diagnosis process include the following parts: knowledge base, inference machines, human-computer interface, diagnostic results, conclusions, and renewal of knowledge. By designing the interaction between use cases, the requirements of the PHM system are met.

4.2.2 Function Models for PHM Systems

The establishment of the functional model belongs to the second stage of the RFLP method. Therefore, the items in the requirements model are disassembled into the relevant execution components at this stage. In the general activity diagram of functional analysis, data is received from the upper-level model. The information is divided into different layers (see Figure 10). Functional models are mainly composed of various activity diagrams. At the beginning of the functional analysis stage, a workflow for the PHM system

has been developed according to the requirement scenario in the requirement analysis stage. Besides, based on the action of the activity diagram, functional decomposition can be carried out to get the detailed functional hierarchy by black box analysis. For example, the fault diagnosis function can be decomposed into feature extraction and diagnosis reasoning functions. Furthermore, feature extraction can be realized based on data fusion and data acquisition. Diagnosis reasoning can be decomposed into fault prediction and diagnosis functions. Correspondingly, a white box activity is used to describe the implementation process of the fault diagnosis function.

The overall activity diagram in the functional model shows the process of self-testing, self-protection, and fault diagnosis for the PHM system, as shown in Figure 11A. The self-testing process includes data acquisition, data transmission, data processing, and

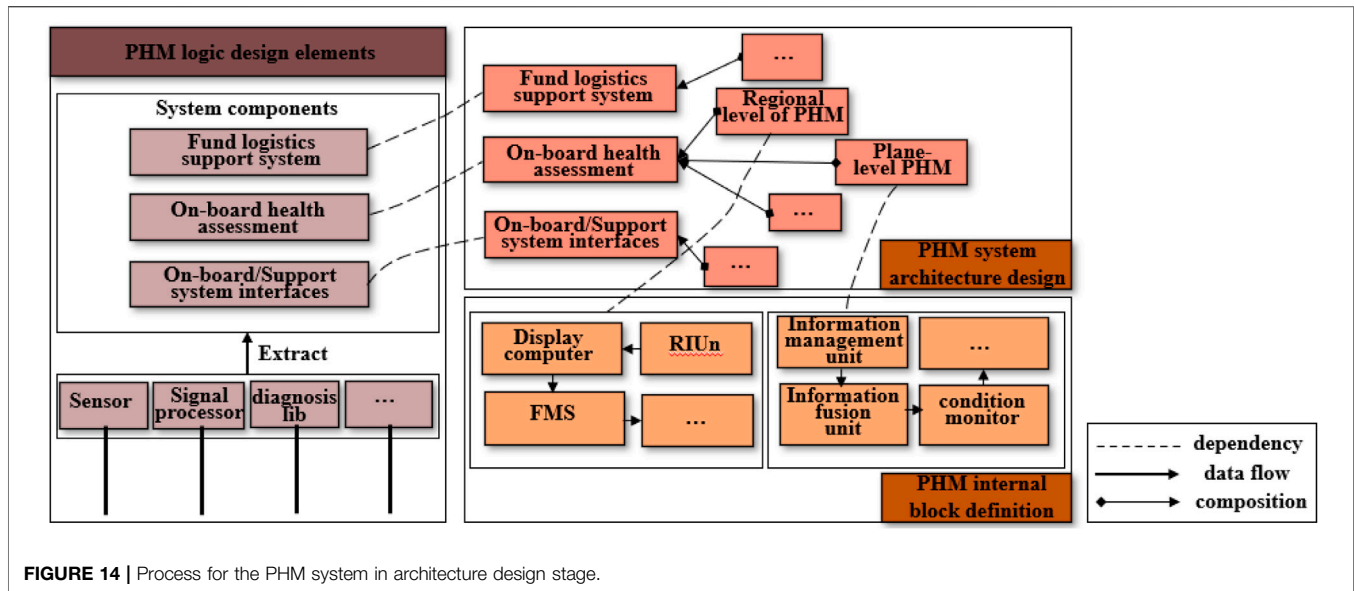


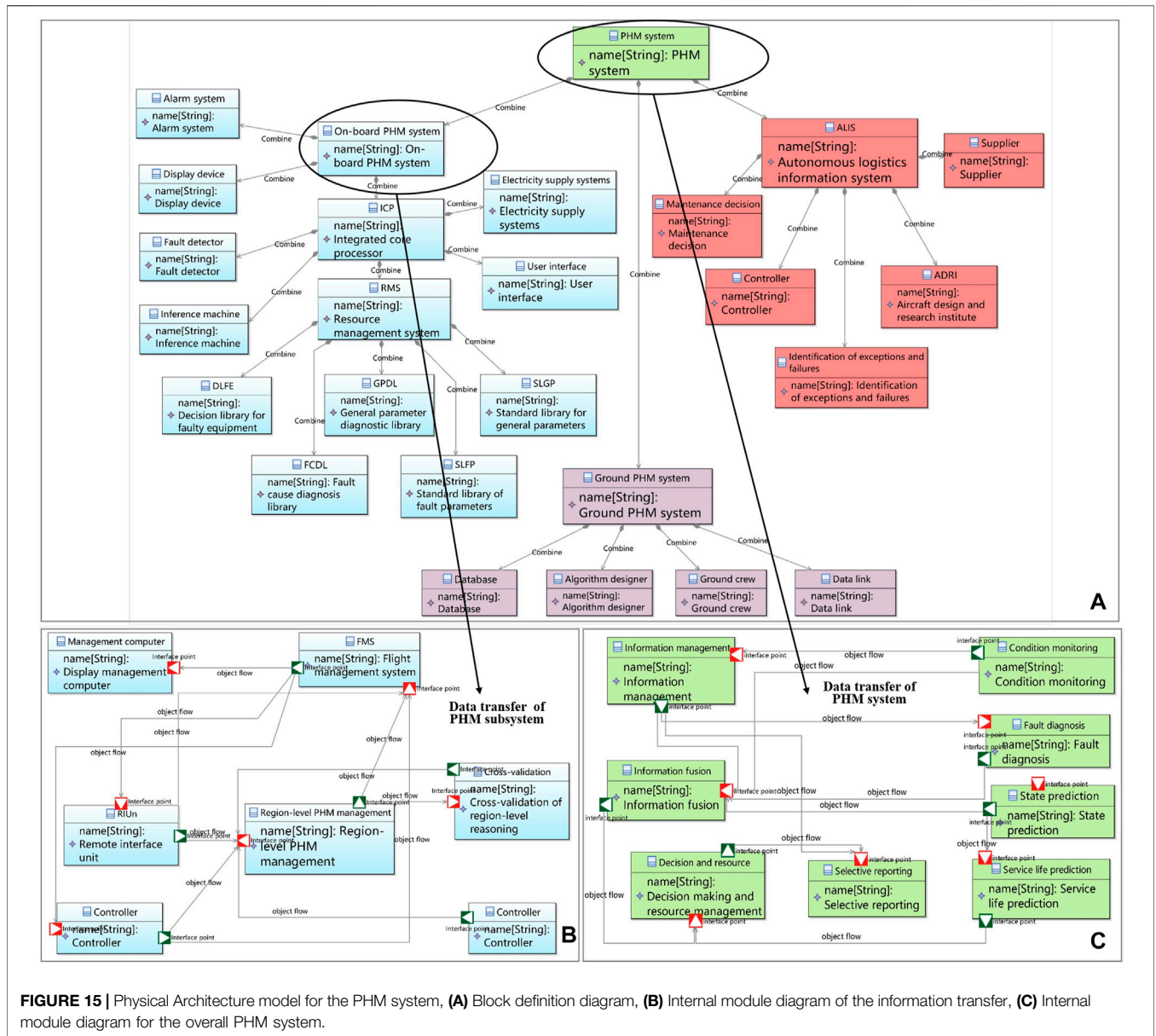
FIGURE 14 | Process for the PHM system in architecture design stage.

status monitoring. The evaluation results obtained from the self-testing process determine the decision-making and fault alarm in the self-protection process. Since it cannot show the complexity of fault diagnosis, it is necessary to add other activity diagrams to supplement the model. As a result, the black box activity diagram (see Figure 11B) and white box activity diagram (see Figure 11C) are added to the model. Black box activity diagrams can ensure that the functions of the PHM system meet the requirements. Then it disintegrates fault diagnosis into characteristic parameter extraction and fault reasoning and isolation. It does not consider the internal logical structure and internal characteristics of the system, and only depends on the description of the upper-level requirements. The black box activity diagrams for fault diagnosis in this case are driven by data. The data mainly includes sensor data, instruction data, fault data, and status data. The white box activity diagram is based on the black box activity diagram and is used to further explain the black box activity diagram. The white-box activity diagrams in this case are driven by logic. All logical paths in the fault detection process are traversed to check the system status from different points. The status parameters mainly include power supply information, valve pressure, fuel tank level, and pipeline temperature. According to the deviation between the actual state and the expected state, fault information and decision information are sent to the display system and the control system, respectively.

4.2.3 Logic Architecture Models for PHM Systems

The self-testing, self-protection, and fault diagnosis of the PHM system need to be supported by logical activities that can transmit information. The logical model is established to ensure that the activities of the functional model can be carried out normally. As shown in Figure 12, the activity diagrams of the functional model are further converted into the logical activities of the sequence diagrams. The activities in the activity diagram correspond to signal or event transfers between lifelines in the sequence diagram. For example, the

activity of sensor signal acquisition can be carried out by the message transfer between sensor and signal transmitter. Based on the sequence diagram, the participants in the activity can be determined by analyzing the interaction logic of the activities. The state changes of actors over time in the sequence diagrams need to be supported by the state machine diagram. In the process of fault diagnosis, it is inevitable to judge the causes of fault. Therefore, fault tree design is introduced into the logical model. Figure 13A is the sequence diagram of the fault diagnosis mechanism. This sequence diagram realizes the logic-driven information transmission of the PHM system: the fault signal obtained from the sensor is processed and sent to the control center to guide the work of the fuel system. Figure 13B is the sequence diagram of the fault diagnosis interaction. This sequence diagram shows data-driven information transfer activities among operation staff, the standard parameter knowledge base, diagnostic parameter knowledge base, inference engine, failure equipment decision base, and operational knowledge base. The transmission of information between these knowledge bases is an essential basis for fault reasoning and isolation. Figure 13C is the state machine diagram illustrating the dynamic behavior of the PHM system. It shows the evolution of the knowledge base, inference machine, and display system over time. The reading, storage, cleaning, and feature extraction of data information in these dynamic behaviors are completed in the state machine diagram. A typical aviation fuel system fault tree is developed to support the fuel valve fault diagnosis as shown in Figure 13D. Through fault tree analysis (FTA), the reasons for the fault top event are obtained as follows: valve control fault, sensor fault, and valve fault. The reasons for valve control faults include valve power failure, valve control panel switch failure, and signal cable failure. The reasons for sensor faults include pressure relay failure, closing abnormal failure, and starting abnormal failure. The reasons for valve fault failure



include one-way valve failure and backup valve failure. By analyzing the fault factors, including hardware, software, environment, and human factors, the FTA can determine the various possible combinations of causes and the occurrence probability.

4.2.4 Physical Architecture Models for the PHM System

The physical architecture model consists of all the physical components of the PHM system. As shown in Figure 14, based on the above analysis, the design of the physical model in this case is based on the tasks of self-testing, self-protection, and fault diagnosis. The data in the logical models of the upper layer is used as the input. The block definition diagrams are designed to show the physical architecture of the PHM system.

All the modules are added to the block definition diagram. The information transfer of each module in the physical architecture is realized by the internal module diagram. Figure 15A is the block definition diagram of the physical architecture, including the software and hardware, as well as interface forms and connection methods. In summary, the physical architecture consists of three PHM subsystems: the on-board PHM system, the ground PHM system, and the autonomous logistics information system. In the physical architecture model, the internal block diagram is a further complement to the block definition diagram. Figure 15B is the internal module diagram of information transfer between physical components of the on-board PHM system. Figure 15C is the internal module diagram of the overall PHM system. The whole working process of the PHM system includes data collection, data processing, status

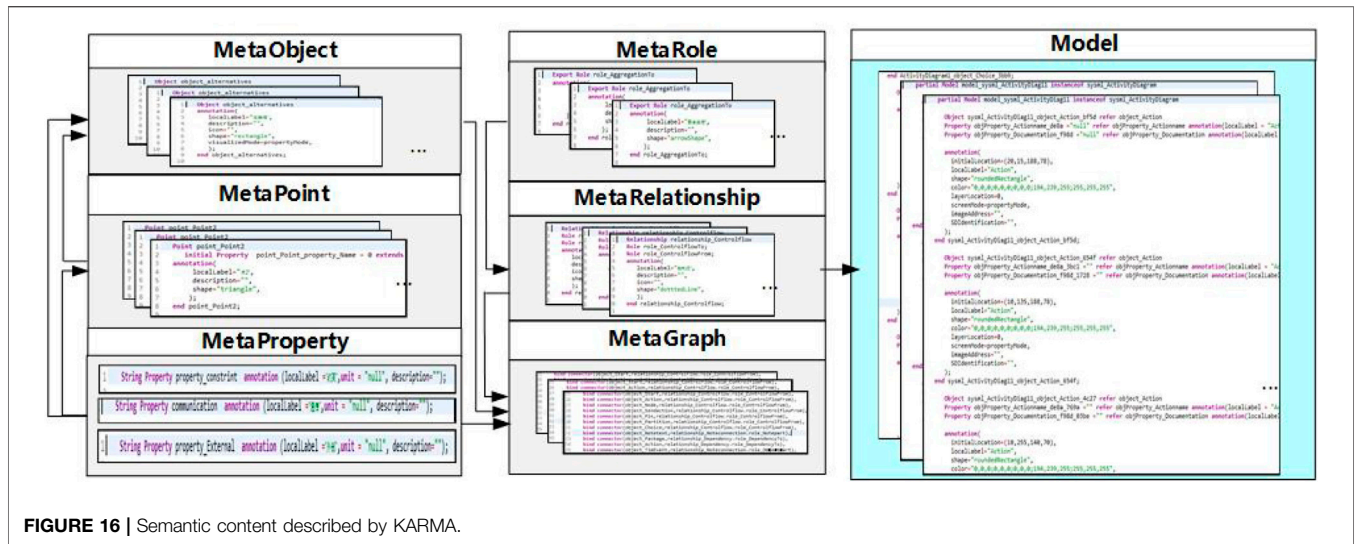


FIGURE 16 | Semantic content described by KARMA.

monitoring, fault detection, health assessment, and health management.

4.2.5 Semantic Modeling and Generated Ontology

Based on the semantic modeling method, the semantic content can be used to describe the graphical models by KARMA. As Figure 16 shows, meta-models and models are described by their respective KARMA files. All the graph contents are expressed by the KARMA language grammar specification. Each graphical model introduced above has a corresponding text model. The text models are generated automatically in MetaGraph 2.0.

After a PHM system design, the design information cannot be effectively managed by graphical models. Although the MBSE models can express the design information clearly during practice, there are still some problems in the management and analysis of the design information, which are listed as follows.

- 1) Shareable capability of design process. The PHM design information contains a large number of concepts from different fields. It is important to contribute to a shareable open source glossary to help designers communicate effectively.
- 2) Quantitative analysis for the system design elements. Quantitative analysis of design information can help designers understand the scale of the designed system better.
- 3) Logical verification for system design. The whole design process of the PHM system aims to achieve user requirements. It is necessary to verify whether the design content covers the earlier requirements.

In order to solve the above problems, an ontology about PHM MBSE design is generated based on semantic modeling, which is introduced as follows.

a. Shareable capability

Based on the transformation rules defined in Section 3.1, all the design information in MBSE models can be extracted and transformed into OWL based on ontology. The ontology is represented in Protégé as shown in Figure 17. The six classes are created in Figure 17–A. The additional project class, connector class, function class, and plugin class are defined to describe the extra information. The relationship between meta models is linked by object property, as shown in Figure 17B. The individuals represent the model elements in the M1 layers of MOF, as shown in Figure 17C. Figure 17D shows the whole structure of the ontology, which intuitively shows the complexity of the ontology. The ontology contains all the design information about the PHM MBSE design, including the maintenance function scenario, valve fault diagnosis process, etc. The construction of an ontology follows a standard specification. As soon as a designer abides by the ontology specification, ontology can be used and developed by stakeholders from different fields. Thus, ontology can be considered as an open source knowledge base which can share the knowledge about PHM system. Besides, when extending the ontology, it is necessary to confirm the concept definition and data consistency. Based on the semantic modeling, the ontology content comes from the system design information. The additional PHM-related concepts can be designed based on the meta-model library. So the concept definition and consistency assurance can be eliminated, which can improve the communication efficiency.

b. Quantitative Analysis

Through the above system design, a total of 21 models have been constructed. Determining the specific numbers of design elements, such as object individuals, graph individuals, and property individuals, is necessary for users to understand the system objectively. Based on the query ability, the MBSE models can be evaluated via ontology. As a commonly used standard RDF query language, SPARQL Protocol and RDF Query Language

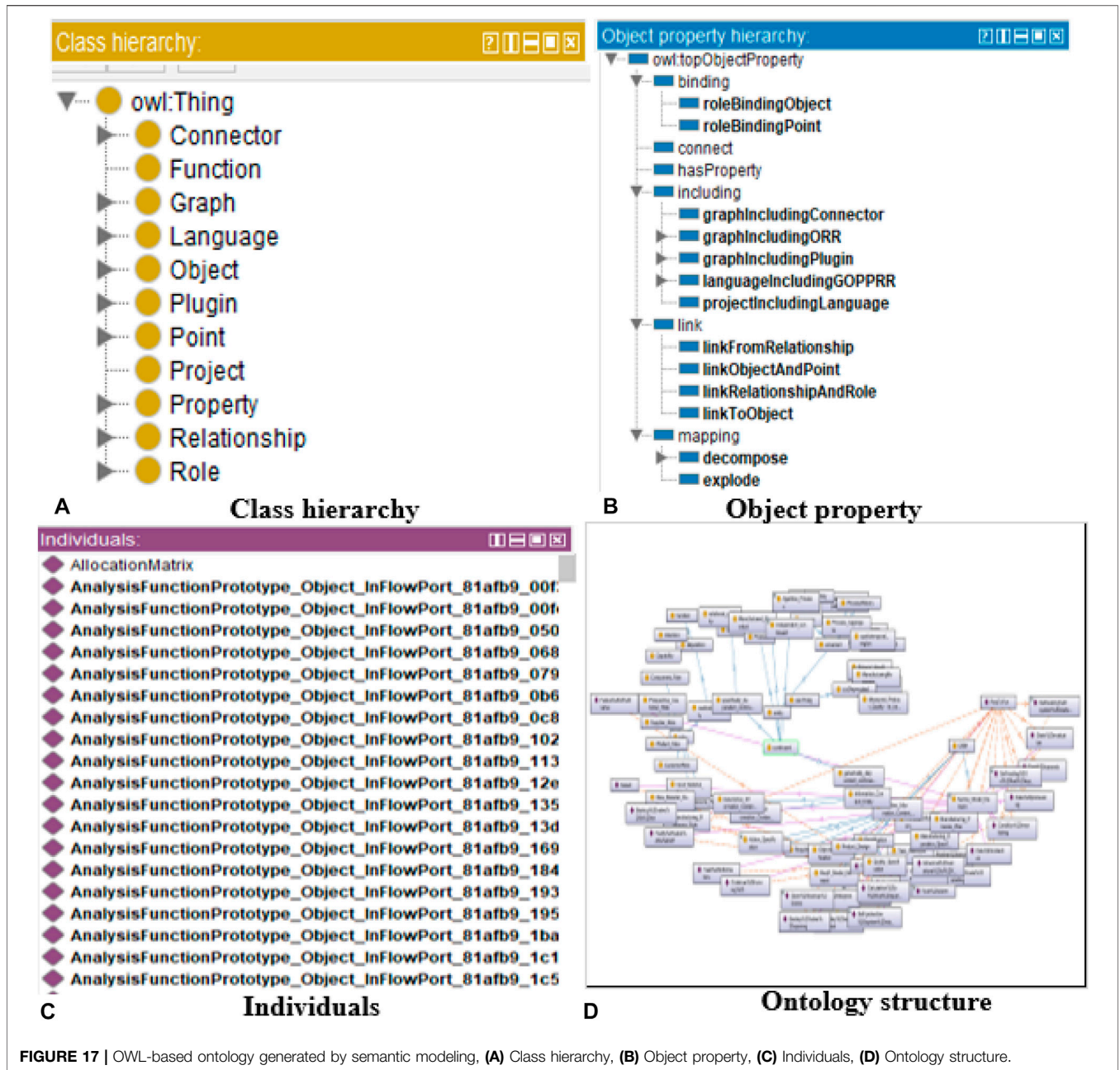


FIGURE 17 | OWL-based ontology generated by semantic modeling, (A) Class hierarchy, (B) Object property, (C) Individuals, (D) Ontology structure.

(SPARQL) are used in this case. In order to identify the components in the PHM system, a query text is developed as shown in Algorithm 3. The SPARQL query text aims to count PHM system design elements in MBSE models. Figure 18B shows the query results, which demonstrates that all the components and their property information are captured. All statistical results are listed in Table 4. The total individual count is 3,121, which is the sum of the other seven kinds of individuals. These results can be used to quantify the system complexity, which can be used as an evaluation indicator to help designers evaluate the PHM system during the system iterative process.

Algorithm 3. KARMA semantic model content

```

1: PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2: PREFIX owl: <http://www.w3.org/2002/07/owl#>
3: PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
4: PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
5: PREFIX se: <http://www.zhoneycomb.com/formats/metagInOwl#>
6:
7: //select all the components information and their properties
8: SELECT ?graph ?componentObject ?name
9: WHERE { ?graph se:graphIncludingObject ?componentObject.
10:         ?componentObject se:hasProperty ?componentNameId;
11:         ?componentNameId se:localLabel ?name
12: }
    
```

c. Logic verification

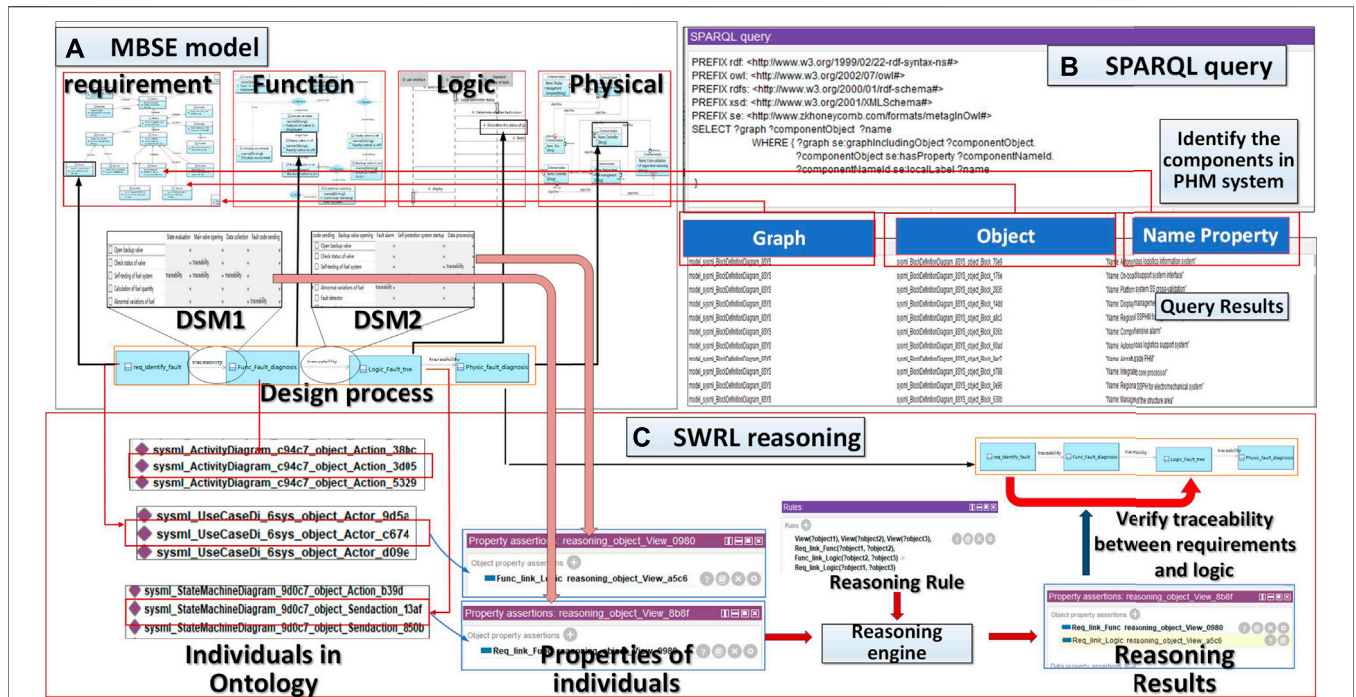


FIGURE 18 | The quantitative analysis and logical reasoning based on ontology, (A) MBSE models for PHM system, (B) SPARQL query, (C) SWRL reasoning.

TABLE 4 | The metrics in the ontology of PHM case.

Metrics	Number	Metrics	Number
individual count	3,121	Graph individual count	21
Object individual count	1,043	Property individual count	973
Point individual count	375	Role individual count	53
Relationship individual count	302	Connector individual count	354

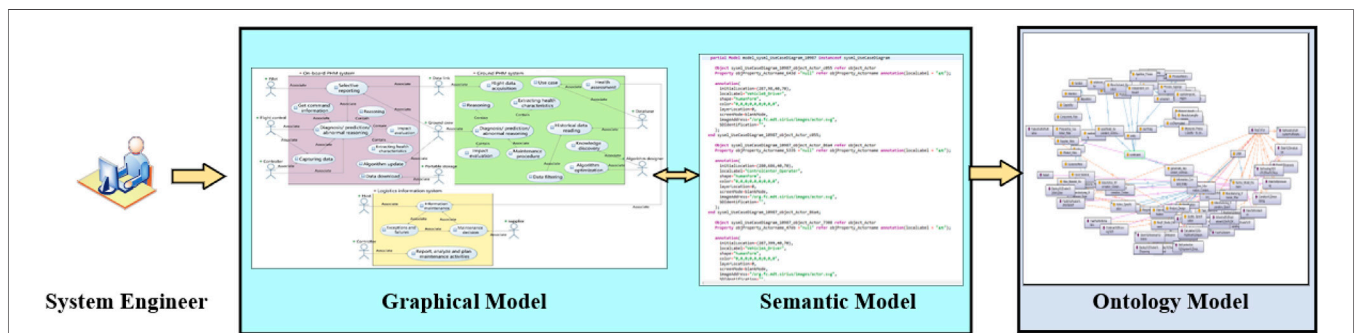


FIGURE 19 | Transformation between the MBSE model and ontology model.

During the system design, it is necessary to verify the requirement traceability at different design stages. In order to determine the traceability, Semantic Web Rule Language (SWRL) is applied to understand the relationship between design elements. As shown in Figure 18A, two DSMs have been developed. One is used to represent the relationships between

requirement models and functional models, and the other is used to represent the relationship between functional models and logic models. A reasoning text is developed and the reasoning algorithm is shown in Algorithm 4. Based on the reasoning rule, a new object property is generated to describe the relationship between requirement information and logical

information. The reasoning process is conducted by the built-in reasoning engine in protégé. The reasoning results are shown in **Figure 18C**. The *sysml_UseCaseDi_6sys_object_Actor_-c674* refers to the requirement that the PHM system must have the self-test function. The *sysml_ActivityDiagram_c94c7_object_Action_3d05* and *sysml_StateMachineDiagram_9d0c7_object_Sendaction_13af* are the design information of functional analysis stage and logical design respectively. A new object property is generated by the reasoning rule. This object property describes the relationship between a requirement and a logic design element. Thus, based on the determination of relationships between different design processes, the cross-viewing design elements can be traced and the logic verification can be realized.

Algorithm 4. SWRL for logic reasoning

```

1: Karma_Model_Element(?DSMelement1)
2: Karma_Model_Element(?DSMelement2)
3: Karma_Model_Element(?DSMelement3)
4: object_Actor(?actor1)
5: object_Action(?action1)
6: object_State(?state1)
7: DSMelementBindingKarmaElement(?DSMelement1,?actor1)
8: DSMelementBindingKarmaElement(?DSMelement2,?action1)
9: DSMelementBindingKarmaElement(?DSMelement3,?state1)
10: transceability(?DSMelement1,?DSMelement2)
11: transceability(?DSMelement2,?DSMelement3) →
12: transceability(?actor1,?state1)

```

In general, as shown in **Figure 19**, the whole process of transformation from MBSE model to ontology has been achieved by semantic modeling. Semantic modeling is able to express the information of a graphical MBSE model completely in text-based files. Besides, ontologies can be generated by semantic modeling. The MBSE models can be analyzed and evaluated by ontology to improve the system's development.

5 CONCLUSION

In this case, a semantic MBSE is proposed to design an aircraft PHM system. As a multidisciplinary system, the design process of the PHM system generates a large amount of information that needs to be analyzed and evaluated. Different with the traditional graphical MBSE model, semantic modeling approach is expected to support the mutual conversion between MBSE model and ontology model. The GOPPRRE concepts based on an M0-M3 modeling framework are used to realize semantic modeling. Based on this framework, a novel MBSE language, KARMA, is applied to complete the MBSE design of the PHM system. The fuel system of an aircraft is selected to verify the feasibility and effectiveness of the proposed semantic MBSE method. According to the results of a case study, the whole process of transformation from MBSE model to ontology has been achieved by semantic modeling. Besides, the design elements of the PHM system can be analyzed and evaluated by the developed ontology. The conclusions of this study are listed as follows.

- A semantic modeling language, KARMA, based on the GOPPRRE approach, has been used to support the PHM system design. The MBSE models constructed based on

KARMA enable the representation of the PHM system design elements.

- A RFLP modeling process is realized to design the PHM system. This applied modeling process starts from requirements analysis to functional analysis, logical architecture design, and physical architecture design.
- A domain-specific meta-model library based on the GOPPRRE approach is constructed to support the PHM system design. In the case study of an aircraft fuel PHM system, the constructed domain-specific meta-model library contains 13 kinds of meta-graph, 56 kinds of meta-object, and 31 kinds of meta-relationship.
- The semantic modeling method based on the GOPPRE has been used to support the description of traditional graphical MBSE models. On this basis, the ontology model can be converted from MBSE models by the semantic modeling method. The case results show that a semantic ontology model can completely represent the MBSE design information.
- The constructed ontology has enabled knowledge sharing in the field of PHM systems. With the query capability, the ontology also enables quantitative analysis of the PHM system scales, including individual count, object count, graph count, and so on. According to the query results, the individual count is 3,121. The object individual count is 1,043, which accounts for the largest proportion of the design elements. This reminds designers of the need to focus on object elements if they want to optimize the system. Besides, traceability between requirement design element and logical design element has been reasoned by the SWRL, which prove the ontology can be applied to the logical verification in product design process.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/Supplementary Material; further inquiries can be directed to the corresponding author.

AUTHOR CONTRIBUTIONS

JC: conceptualization, methodology, writing–original draft; YC: software, data curation, writing–original draft; ZH: software, formal analysis, visualization; JL: software, formal analysis; XZ: writing–review, formal analysis; HZ: writing–review; DK: supervision.

FUNDING

This work was supported by the National Fundamental Research Project (Grant No. JCKY2020208B036, MKF20200020). Open access funding provided by the Swiss Federal Institute of Technology, Lausanne

REFERENCES

- Allen, J. L. (2016). "An Overview of Model-Based Development Verification/validation Processes and Technologies in the Aerospace Industry," in Proceeding of the AIAA Modeling and Simulation Technologies Conference, San Diego, California, USA, 4-8 January 2016. doi:10.2514/6.2016-1922
- Bikakis, N., Tsinaraki, C., Gioldasis, N., Stavrakantonakis, I., and Christodoulakis, S. (2013). "The Xml and Semantic Web Worlds: Technologies, Interoperability and Integration: a Survey of the State of the Art," in *Semantic Hyper/multimedia Adaptation* (Springer), 319–360. doi:10.1007/978-3-642-28977-4_12
- Bindschadler, D. L., Smith, R. R., Valerio, C. P., and Schimmels, K. A. (2017). "A Structured, Model-Based System Engineering Methodology for Operations System Design," in *Space Operations: Contributions from the Global Community* (Springer), 271–290. doi:10.1007/978-3-319-51941-8_13
- Che, C., Wang, H., Ni, X., and Hong, J. (2018). Fault Fusion Diagnosis of Aero-Engine Based on Deep Learning. *J. Beijing Univ. Aeronautics Astronautics* 44, 621–628. doi:10.13700/j.bh.1001-5965.2017.0197
- Cheng, J., Ma, Z., and Tong, Q. (2016). "Rdf Storage and Querying: a Literature Review," in *Handbook of Research on Innovative Database Query Processing Techniques* (Hershey, PA: IGI Global), 460–478.
- Chinosi, M., and Trombetta, A. (2012). Bpmn: An Introduction to the Standard. *Comput. Stand. Interfaces* 34, 124–134. doi:10.1016/j.csi.2011.06.002
- Deng, N. (2018). What Is Temporal Ontology? *Philos. Stud.* 175, 793–807. doi:10.1007/s11098-017-0893-6
- Dumargue, T., Pougeon, J. R., and Massé, J. (2016). "An Approach to Designing Phm Systems with Systems Engineering," in Proceeding of the Third European Conference of the Prognostics and Health Management Society 2016, July 5–8, 2016 (Bilbao: PHMSociety).
- Estefan, J. A. (2007). Survey of Model-Based Systems Engineering (Mbse) Methodologies. *IncoSE MBSE Focus Group* 25, 1–12.
- Feiler, P. H., Gluch, D. P., and Hudak, J. J. (2006). *The Architecture Analysis & Design Language (AADL): An introduction* Tech. Rep. Carnegie Mellon University.
- Friedenthal, S., Griego, R., and Sampson, M. (2007). "IncoSE Model Based Systems Engineering (Mbse) Initiative," in *INCOSE International Symposium* (San Diego, CA: INCOSE).
- Friedenthal, S., Moore, A., and Steiner, R. (2014). *A Practical Guide to SysML: The Systems Modeling Language (Morgan Kaufmann)*. Burlington: Morgan Kaufmann.
- Gray, J., and Rumpe, B. (2021). Reflections on the Standardization of Sysml 2. *Softw. Syst. Model* 20, 287–289. doi:10.1007/s10270-021-00881-2
- Ha, I., Oh, K.-J., Hong, M.-D., Lee, Y.-H., Rosli, A. N., and Jo, G.-S. (2014). Ontology-driven Visualization System for Semantic Searching. *Multimed. Tools Appl.* 71, 947–965. doi:10.1007/s11042-011-0889-8
- Hahm, G. J., Lee, J. H., and Suh, H. W. (2015). Semantic Relation Based Personalized Ranking Approach for Engineering Document Retrieval. *Adv. Eng. Inf.* 29, 366–379. doi:10.1016/j.aei.2015.01.003
- Hause, M., Brookshier, D., and Bleakley, G. (2012). *UPDM-unified Profile For DoDAF/MODAF*. Tech. Rep. Washington, DC: Department Of Defense.
- Hu, Y., Miao, X., Si, Y., Pan, E., and Zio, E. (2022). Prognostics and Health Management: A Review from the Perspectives of Design, Development and Decision. *Reliab. Eng. Syst. Saf.* 217, 108063. doi:10.1016/j.ress.2021.108063
- Kalgren, P. W., Byington, C. S., and Roemer, M. J. (2006). "Defining Phm, a Lexical Evolution of Maintenance and Logistics," in Proceeding of the 2006 IEEE Autotestcon, Anaheim, CA, USA, 18-21 Sept. 2006. doi:10.1109/autest.2006.283685
- Kelly, S., and Tolvanen, J.-P. (2008). *Domain-specific Modeling: Enabling Full Code Generation*. John Wiley & Sons.
- Kulkarni, C. S., Roychoudhury, I., and Schumann, J. (2018). "On-board Battery Monitoring and Prognostics for Electric-Propulsion Aircraft," in *2018 AIAA/IEEE Electric Aircraft Technologies Symposium* (Cincinnati, Ohio: AIAA). doi:10.2514/6.2018-5034
- Liao, L., and Köttig, F. (2016). A Hybrid Framework Combining Data-Driven and Model-Based Methods for System Remaining Useful Life Prediction. *Appl. Soft Comput.* 44, 191–199. doi:10.1016/j.asoc.2016.03.013
- Lu, J., Ma, J., Zheng, X., Wang, G., Li, H., and Kiritsis, D. (2021). Design Ontology Supporting Model-Based Systems Engineering Formalisms. *IEEE Syst. J.* 14, 1–12. doi:10.1109/jsyst.2021.3106195
- McGuinness, D. L., and Van Harmelen, F. (2004). Owl Web Ontology Language Overview. *W3C recommendation* Available at: www.w3.org/TR/owl-features.
- Meersman, R. A. (1999). "Semantic Ontology Tools in Is Design," in *International Symposium on Methodologies for Intelligent Systems* (Springer), 30–45. doi:10.1007/bfb0095088
- Pourbabae, B., Meskin, N., and Khorasani, K. (2016). Sensor Fault Detection, Isolation, and Identification Using Multiple-Model-Based Hybrid Kalman Filter for Gas Turbine Engines. *IEEE Trans. Contr. Syst. Technol.* 24, 1184–1200. doi:10.1109/tcst.2015.2480003
- Roques, P. (2017). *Systems Architecture Modeling with the Arcadia Method: A Practical Guide to Capella*. Elsevier.
- Song, Y., Liu, D., Yang, C., and Peng, Y. (2017). Data-driven Hybrid Remaining Useful Life Estimation Approach for Spacecraft Lithium-Ion Battery. *Microelectron. Reliab.* 75, 142–153. doi:10.1016/j.microrel.2017.06.045
- Sutrisno, E., Oh, H., Vasan, A., and Pecht, M. (2012). "Estimation of Remaining Useful Life of Ball Bearings Using Data Driven Methodologies," in Proceeding of the 2012 IEEE Conference on Prognostics and Health Management, Denver, CO, USA, 18-21 June 2012 (IEEE). doi:10.1109/icphm.2012.6299548
- Torre, D., Labiche, Y., Genero, M., and Elaasar, M. (2018). A Systematic Identification of Consistency Rules for Uml Diagrams. *J. Syst. Softw.* 144, 121–142. doi:10.1016/j.jss.2018.06.029
- Wang, H., Wang, G., Lu, J., and Ma, C. (2019). "Ontology Supporting Model-Based Systems Engineering Based On a GOPPRR Approach (New Knowledge in Information Systems and Technologies)," in *New Knowledge in Information Systems and Technologies* (Galicia: Springer), 426–436. doi:10.1007/978-3-030-16181-1_40
- Wang, L., Izygon, M., Okon, S., Wagner, H., and Garner, L. (2016). "Effort to Accelerate Mbse Adoption and Usage at Jsc," in Proceeding of the AIAA SPACE 2016, Long Beach, California, 13 - 16 September 2016 (IEEE), 5542. doi:10.2514/6.2016-5542
- Xu, J., Wang, Y., and Xu, L. (2014). Phm-oriented Integrated Fusion Prognostics for Aircraft Engines Based on Sensor Data. *IEEE Sensors J.* 14, 1124–1132. doi:10.1109/jssen.2013.2293517
- Yang, H., Baraldi, P., Maio, F. D., and Zio, E. (2015). Online Performance Assessment Method for a Model-Based Prognostic Approach. *IEEE Trans. Reliab.* 65, 781–735. doi:10.1109/TR.2015.2500681
- Ye, Q., Shen, S., and Ma, L. (2020). "Research on the Model Integration in Aircraft Phm Development," in *2020 Global Reliability and Prognostics and Health Management (PHM-Shanghai)* (Shanghai: IEEE). doi:10.1109/phm-shanghai49105.2020.9280992
- Zacharewicz, G., Daclin, N., Doumeings, G., and Haidar, H. (2020). Model Driven Interoperability for System Engineering. *Modelling* 1, 94–121. doi:10.3390/modelling1020007
- Zaidan, M. A., Harrison, R. F., Mills, A. R., and Fleming, P. J. (2015). Bayesian Hierarchical Models for Aerospace Gas Turbine Engine Prognostics. *Expert Syst. Appl.* 42, 539–553. doi:10.1016/j.eswa.2014.08.007
- Zhao, Y., Li, Z., Wang, Z., Xu, R., and Ding, E. (2022). "Fault-tolerant Center of Gravity Control for Fuel Systems with Component Failures," in *Advances in Guidance, Navigation and Control* (Springer), 4327–4336. doi:10.1007/978-981-15-8155-7_359
- Zhao, Z., Bin Liang, B., Wang, X., and Lu, W. (2017). Remaining Useful Life Prediction of Aircraft Engine Based on Degradation Pattern Learning. *Reliab. Eng. Syst. Saf.* 164, 74–83. doi:10.1016/j.ress.2017.02.007
- Zheng, X., Lu, J., and Kiritsis, D. (2021). The Emergence of Cognitive Digital Twin: Vision, Challenges and Opportunities. *Int. J. Prod. Res.*, 1–23. doi:10.1080/00207543.2021.2014591

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Copyright © 2022 Chen, Chen, Hu, Lu, Zheng, Zhang and Kiritsis. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.