Reliability Engineering and System Safety 193 (2020) 106598

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress

A systematic methodology for Prognostic and Health Management system architecture definition

Rui Li*, Wim J.C. Verhagen, Richard Curran

Delft University of Technology, Delft, 2629HS, the Netherlands

A R T I C L E I N F O

Keywords:
Prognostic and Health Management (PHM)
System architecture
System engineering (SE)
Aircraft maintenance

A B S T R A C T

Prognostic and Health Management (PHM) systems support aircraft maintenance through the provision of diagnostic and prognostic capabilities, leveraging the increased availability of sensor data on modern aircraft. Diagnostics provide the functionalities of failure detection and isolation, whereas prognostics can predict the remaining useful life (RUL) of the system. In literature, PHM technologies have been studied from different perspectives, covering various aims such as improving aircraft system reliability, availability, safety and reducing the maintenance cost. From a design perspective, several conceptual formulations of design methodologies are available, enabling a set of PHM system architectures based on different frameworks and the derivation of system requirements. However, a systematic methodology towards a consistent definition of PHM architectures has not been well established. The characteristics of architectures have not been dealt with in depth. To address these gaps, this paper presents a systematic methodology for PHM architecture definition to ensure a more complete and consistent design during the development phase of the product lifecycle. Moreover, a generic PHM architecture in accordance with this systematic methodology is proposed in this article. A case study is conducted to verify and validate the architecture, ensuring it meets the requirements for a correct and complete representation of PHM characteristics.

1. Introduction

Prognostics and health management (PHM) has emerged as one of the key solutions for improving system reliability, safety, maintainability, supportability, and economic affordability for major industrial assets (e.g. aircraft, power plants, trains). A growing amount of literature has evaluated diagnostic and prognostic technologies with the aim to optimize asset operations and maintenance while improving safety, reliability, and cost-effectiveness [1,2]. Moreover, many papers discuss key aspects of system maintenance and PHM systems, such as maintenance principles [3–5], cost and efficiency [6–8], safety and reliability [9].

PHM describes a set of capabilities involving both diagnostics and prognostics: diagnostics concerns the process of fault detection and isolation, while prognostics is the process of predicting the future state or remaining useful life (RUL) according to current or historical conditions [10]. In Niu's research [11], it is stated that the design team should have a thorough understanding of methods for optimal selection of monitoring strategies, tools, and algorithms needed to detect, isolate, and predict the time evolution of the fault, as well as systems, approaches for designing experiments and testing protocols, performance metrics, and means to verify and validate the effectiveness and performance of the selected models. PHM research has a specific focus towards the management of some of this complexity via monitoring, diagnostic, and prognostic technologies. The strategic application of PHM technologies has been shown to effectively reduce equipment/process downtime and lower maintenance costs [12]. Part of the challenge of PHM, particularly for manufacturers, is to know exactly how to apply PHM within the operations to gain the maximum actionable information [12].

Currently, in research a number of applications have been developed for asset-specific modeling and prediction in an independent fashion [13–15]. Consequently, there is some inconsistency in the understanding of key concepts for designing prognostic systems. In order to progress from application-specific solutions towards structured, consistent and efficient PHM system implementations, the development and/or use of suitable methodology is essential [16]. Such a methodology should address the following high-level requirements: 1) it should be *unambiguous*, i.e., the concepts and terminology used should be defined well, without being open to multiple competing interpretations; 2) it should be *comprehensive,* i.e., it should cover all essential steps in developing a PHM system; 3) it should be *pragmatic,* i.e.,

---

* Corresponding author.

E-mail addresses: R.Li-4@tudelft.nl (R. Li), W.J.C.Verhagen@tudelft.nl (W.J.C. Verhagen), R.Curran@tudelft.nl (R. Curran).

https://doi.org/10.1016/j.ress.2019.106598
Received 13 December 2018; Received in revised form 27 June 2019; Accepted 4 August 2019

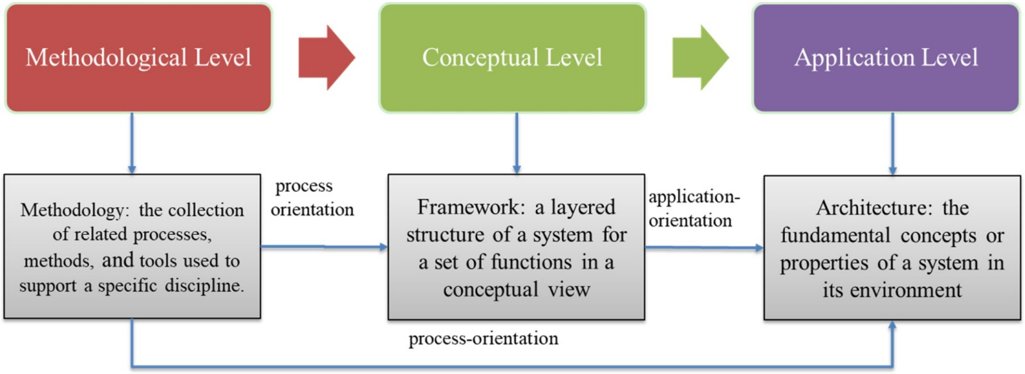

Fig. 1. Key terms and their interrelations.

researchers and practitioners alike should be able to apply the methodology in a straightforward fashion. This paper addresses all elements identified.

In relation to point 1, it is essential to identify and define the following three key terms and their interrelations: methodology, framework, and architecture. The definitions and their interpretation are identified in Fig. 1. Here, methodology is viewed from the lens of design, where the concept of design methodology refers to the development or method for a unique situation, with the collection of related processes, methods, and tools used to support a specific discipline [17]. As such, a methodology does not provide solutions – rather, it is the systematic study of approaches to generate solutions. Moving one step from process to actual ideation and instantiation, the term ‘framework’ mainly describes the layered structure of a system for a set of functions in a conceptual view. Building on this conceptual perspective, the system architecture moves to the application level and concerns the fundamental concepts or properties of a system in its environment as embodied in its elements, relationships, and in the principles of its design and evolution [18]. The concept of ‘view’ is important to mention in relation to the system architecture. A view is a representation of a system from the perspective of a related set of concerns, and usually, it is a work product that presents specific architectural data for a given system. A view allows a user to examine a portion of a particular interest area. For example, an Information view may present all functions, organizations, technology, etc. that use a particular piece of information, while the organizational view may present all functions, technology, and information of concern to a particular organization.

Given the complexity and breadth of PHM systems and associated technologies, methodologies are necessary to initiate, sustain and complete PHM system development, thereby covering the conceptual and application levels as mentioned above. Previous research touches upon one or more of these key terms as described in more detail below:

a) Design methodologies

In existing research, a group of authors has reviewed design methodologies for PHM systems and associated techniques [19,20]. For example, Dumargue et al. present various aspects of a design methodology, including general system design, project management considerations, and transversal methodological items, such as model-based systems engineering and methods to manage the technical elements of the system [21]. Cocheteux et al. express a methodology to formalize functional and dysfunctional system knowledge and provided guidelines for designing prognostic process, including a selection of failure modes and associated prognostic tools [22]. Vogl et al. have introduced a process of PHM system development [18]. This process starts with cost and dependability analyses, and then the data management system is initialized for collection, processing, visualization, and archiving of maintenance data. Once the measurement techniques are established, the diagnostic and prognostic approaches are developed and tested. However, this process lacks discussion of the conceptual and application levels; notably, a process of developing a system architecture is missing. Aizpurua et al. [16] formulate a methodology for designing prognostic applications (ADEPS), which is a design selection framework to guide the engineer towards a prognostic approach through a cause-effect flowchart [23]. This research primarily addresses the critical step of selecting and applying an appropriate prognostic approach for PHM applications, but does not PHM cover system development or engineering [24,25].

b) Frameworks

A substantial amount of research has been performed with respect to PHM frameworks. For instance, Mao et al. present a visual model-based framework to simulate and evaluate cloud-based PHM systems [26]. The framework proposes a three-abstraction-layer hierarchical architecture to represent distributed data sources and a cloud-based PHM service center. The design of the framework is based on system modeling language and allows flexible implementations of functional modules and algorithms. Similarly, Yang et al. introduce a new framework on the basis of the concept of a PHM big data center and discuss the associated key technologies, scientific problems and application systems [27]. Zhang et al. recommend a framework integrating health status monitoring and health management of aircraft in order to build a suitable mechanism for managing diagnostics, prediction, and intelligent maintenance decision making [28]. As a common thread, the aforementioned papers propose PHM frameworks in a high-level manner, without a detailed description of elements and interfaces, and without an explicit connection with a governing design methodology.

c) Architecture

The architecture definition for a PHM system plays a critical role to move from a conceptual to an applied level regarding the functions of diagnostics, prognostics and predictive maintenance services for complex assets. A PHM architecture should be complete and consistent over time. To ensure this, updates for knowledge bases and algorithms should be supported, providing an advantage over static legacy systems. For aircraft, one best practice is to develop the on-board and off-board system together [18]. Alternatively, a separate off-board (ground-based) system can be developed, integrating the required diagnostic and prognostic techniques. To meet multidisciplinary requirements in PHM, Han et al. [29] define a distributed and universal platform for the implementation and verification of PHM systems using a configurable system of systems (SoS) architecture. Keller et al. [30] describe the concepts and properties of an onboard HM architecture for aerospace vehicles and how this architecture addresses affordability and can be adapted for a range of aerospace vehicles. Keller et al.’s research also provides a discussion of HM architecture aspects, such as the choice between distributed or centralized, open or proprietary, and flight critical or support critical alternatives, as well as feasible approaches to integrate the related HM functions with an existing system. Towards the use of open system architectures, PHM designers can apply various standards. For instance, the standard ISO-13374 defines an Open System Architecture for Condition-Based Maintenance (OSA-CBM) specification as a standard for moving information in a CBM system [31]. This open architecture provides guidance towards PHM design and enables the interoperability and communication between different CBM systems [32,33]. In addition, IEEE standard 1856 [34] provides information for the implementation of PHM, which can be used by manufacturers and end users for planning implementation and the associated life-cycle operations for the system of interest. However, the architecture of OSA-CBM lacks connection with the higher-level methodology and framework. Also, it lacks to provide detailed application cases of integrating aviation health management systems into the supporting infrastructure for aircraft maintenance.

In summary, existing literature addresses aspects of PHM design methodology and provides PHM architecture formulations. However, a systematic methodology towards a consistent definition of PHM architectures, i.e., one that spans the conceptual and application level, has not been well established. The characteristics of generic PHM architectures have not been dealt with in an in-depth and complete manner; usually, interoperation between PHM system and the aircraft on-bound maintenance/health management systems is lacking. As PHM systems are complex, the design of these systems and their components require the use of systems engineering methods to ensure a more complete and consistent design to mitigate possible rework and ineffectiveness issues during the development life cycle [21]. With these considerations in mind, this paper defines a systematic methodology incorporating functional, logical, and physical views for system architecture definition using a systems engineering approach. Systems engineering provides the methods and tools to design the right product (satisfying customer needs) and design the product right (functional and effective) while optimizing project aspects (quality, cost, time). In addition, a second contribution to the current state of the art is made by proposing, a generic PHM architecture is proposed, incorporating a framework, functional decomposition, functional/logical architecture description, and physical architecture.

The remainder of this article is structured as follows: Section 2 uses a systems engineering approach to propose a systematic methodology for PHM framework and architecture definition. In Section 3, a generic PHM architecture is formulated according to the systemic methodology. Section 4 presents a case study in which the proposed PHM architecture is modeled in SysML to subsequently verify and validate the PHM architecture. Another case study is conducted to demonstrate the consistency, applicability, and compatibility of the PHM architecture through the methods of functions analysis, interfaces analysis, traceability analysis and compliance analysis in Section 5. Finally, conclusions and recommendations for future research are addressed in Section 6.

2. Architecture definition methodology

This section introduces a design methodology for architecture definition using a systems engineering approach, and the novelty of this process is:

- Combining the concept of requirements, functional, logical and physical architectures (“RFLP”) into a PHM architecture design methodology, where the concept of “RFLP” is defined in Section 2.1 and Fig. 2;
- Proposing a systematic PHM architecture design methodology, as highlighted in Section 2.1 and Fig. 3;

These aspects provide a guide for system designers toward the development of PHM architectures in a systematic way. The specifics are addressed in the following sub-sections.

2.1. Architecture definition process

System architecture design has features, properties and characteristics satisfying the problem or opportunity expressed by a set of system requirements (traceable to stakeholder requirements) and life cycle concepts (e.g., operations, support). Architectures are implementable through technologies (e.g., mechanics, electronics, hydraulics, software, services, procedures) [35]. To conduct the architecture definition, this research introduces a methodology, based on the concept of “RFLP” (requirement, functional, logical, and physical architectures), as illustrated in

Fig. 2 [36,37]. This methodology can progress from system requirements, representing the problem from the stakeholders' point of view, as independent of technology as possible, to an intermediate representation of functional/logical architecture, to a subsequent allocation of the functional elements to system elements of a candidate physical architecture, which is related to technologies and is an input of the design solution process [35].

More specifically, this research proposes a methodology operating in a recursive and iterative manner from a system engineering perspective. The methodology flowchart is shown in Fig. 3. Generally, it assumes that system requirements and constraints are available as input in this process. It subsequently incorporates the following primary activities:

1) Task 1: Define system framework;
2) Task 2: Develop the system architecture (functional, logical and physical views);
3) Task 3: Allocate requirements to architecture elements to form derived requirements.

Additionally, the system requirements and related project or technical constraints delivered from the requirements definition process are the inputs of the architecture definition process. The output is the system architecture specification with the traceability information (history, parent requirements, derived requirements, etc.) for each item. Obviously, Task 2 plays a crucial role in constructing a PHM architecture in details, which are highlighted in Fig. 3. Comparing with other methodologies, this task has the novelty of defining architecture from functional, logical and physical views.

2.1.1. Task 1: define system framework

At first, the definition of architecture requires the necessary inputs of system requirements specification as the developed architecture should be fully compliant with requirements, as shown in Fig. 3. While, in some cases, some constraints should be considered in the process of architecture definition, because that may impact the selection and configurations of the technique. Secondly, this process starts with a definition of the system framework. The system framework incorporates the basic structure of a system according to the requirements while comprehending the functions, performance, operational conditions, and project constraints that will influence the architecture [35]. In this case, the associated framework to assist the system architecture development is established and identified according to the defined set of stakeholders' expectations and requirements. Additionally, the system framework describes the layered structure indicating what kind of programs can or should be built and how they would interrelate, as the prototype of architecture, which is on the basis of the current technologies, legacy research, and the knowledge of system [38].

2.1.2. Task 2: develop system architecture

The next activity is to develop the system architecture in view of the

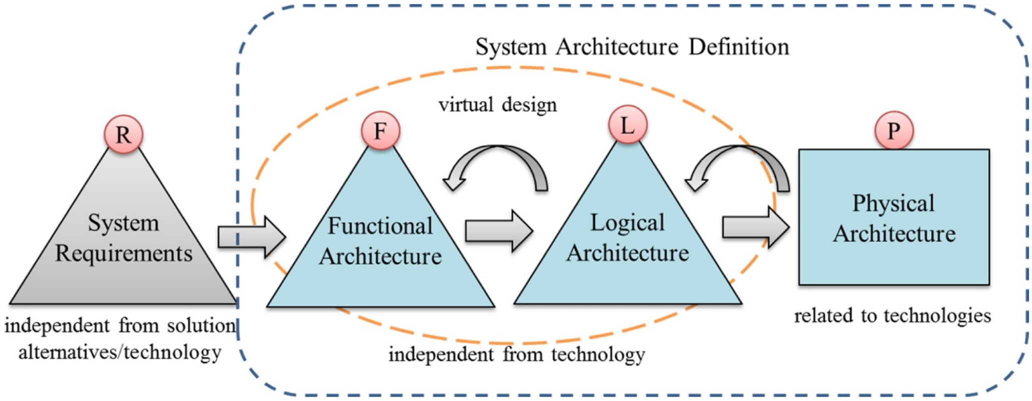

Fig. 2. Concept of “RFLP” in architecture design.

established system high-level framework. The essential aspects of this task are to generate the functional, logical and physical elements and identify the interactions among systems and elements, to complete the design of a system architecture specification compliant with the given requirements and previously defined system framework. As presented in Fig. 3, this task includes the following steps:

a) Task 2.1: Perform functional analysis and functions decomposition

A system is intended to satisfy predefined functions, with the top-level functions defined as the stakeholder need, and a function is a characteristic action or activity that needs to be performed to achieve the desired system objective (or stakeholder need) [38]. Therefore, this task starts with the definition of top-level functions according to the stakeholders’ expectations and intended system objectives. Afterward, these top-level functions are functionally decomposed to lower levels in a hierarchical structure. The process of functions decomposition may consider domain knowledge (e.g. aircraft, PHM, power plant, etc.), the availability of techniques or material (e.g. diagnostic, prognostic, data processing techniques, avionics, monitoring sensors, etc.), as well as the project mission (project objectives, resources, etc.). The functional analysis method can be used to identify and check the functions and sub-functions that accomplish the project mission.

a) Task 2.2: Develop functional architecture

According to the previous steps, sufficient information is available to start the development of functional architecture including system

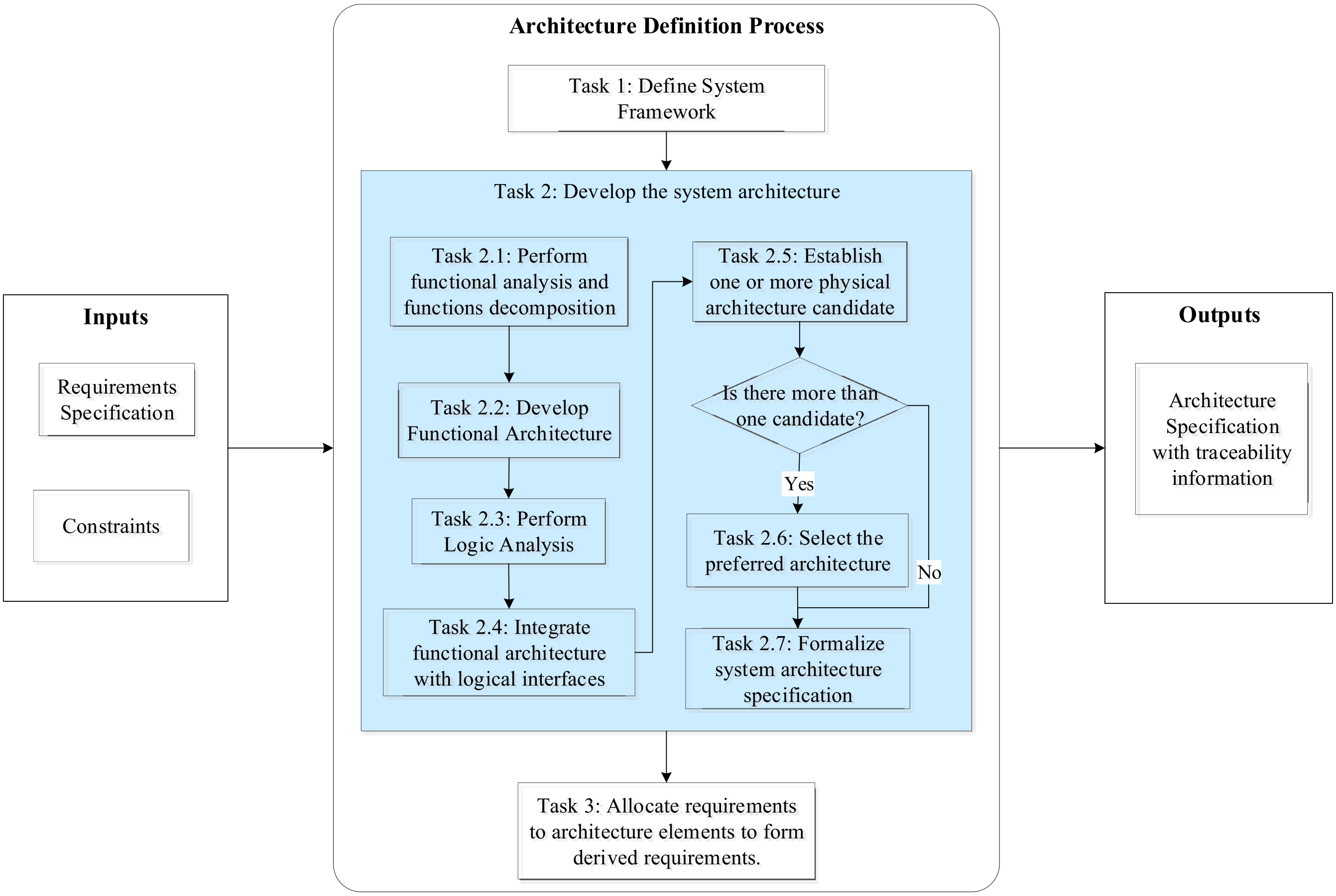

Fig. 3. Architecture definition process.

boundary, functional elements, and external/internal interfaces. In addition to the list of functions from step a), the functional architecture development involves analysis of the functions' hierarchy, input-output flows, and operational scenarios of the target system. In other words, the functional architecture is a set of functions and their sub-functions that enables to identify functional interfaces and interactions between system elements. It ensures that the system functions and the related requirements are analyzed, decomposed, and functionally detailed across the entire system in a feasible and effective manner [39]. Therefore, it can be described with a hierarchical arrangement of elements and interfaces that represent the complete system from a performance and functional perspective in the views of a context or a visual model through the commercial tools, such as Enterprise Architecture [40], CATIA V6 [41] and Rational Rhapsody [42].

a) Task 2.3: Perform logic analysis

The logical architecture is composed of a set of related technical concepts and principles that represent the logical operation of the system. Logic analysis is performed to capture system behaviors, execution sequencing, conditions for control or data-flow, states and operation mode, as well as performance level(s) necessary to satisfy the system requirements [38]. Simultaneously, the trigger condition of the states or operation mode transmission should be identified. The trigger (a control flow) is an element that activates a function as a condition of its execution, which characterizes the logical relationship between different functions or services.

a) Task 2.4: Integrate functional architecture with logic interaction.

Comparing with functional architecture, logical architecture is a structural design that gives as much detail as possible without constraining the architecture to a particular technology or environment [35]. It is the manner in which logical components of a solution are organized and integrated, with the aims of planning and communicating architecture. Both functional and logical architectures are part of a virtual design process. This activity is used to ensure the consistency between functional elements and the sets of logical behaviors.

a) Task 2.5: Establish one or more physical architecture candidates

Afterward, one or more physical architecture candidates are established to determine the elements that can perform system functions and organize them into a physical architecture [43]. Generally, the more candidates there are, the higher the cost will be for evaluation and selection. Due to this consideration, and to maintain effectiveness and economically sensible decision making in practice, the number of candidates is not recommended to exceed 3. In this sense, the physical elements could be materials and artifacts, such as equipment made of hardware, software and/or human roles. Practically, one requirement is that each physical architecture candidate should be compliant with the functional and logical views as well as system requirements via the implementation of related technologies. Hence, the physical elements (configuration item) and interfaces (data flow and format) are specifically identified in each physical architecture candidate.

a) Task 2.6: Select the preferred architecture, if necessary

Once the physical architecture candidates are established, if there is more than one, the preferred one should be selected throughout a trade-off process involving all candidates. Otherwise, this step is skipped [35,44]. It is critical to define how to evaluate the physical architecture candidates; therefore, it is required to establish guiding principles for the system design and evolution metrics, including the list of criteria items (e.g. cost, technical risk, re-usability, economic, pollution, noise) and the criteria weights, which depend on the stakeholders' expectations and project constraints. In the other words, the objective of this task is to provide the "preferred" possible architecture made of suitable system elements and interfaces, that is, the architecture that answers, all the stakeholders' needs and system requirements [44]. The process involves the creation of several candidates; analyzing and assessing the defined candidates by applying system analysis, measurement, and risk management process using the evaluation criteria; as a result, selecting the most suitable one. Moreover, the trade-off concerns the decision making actions that select a solution from various alternatives on the basis of the defined evaluation criteria.

Sometimes, the physical architecture candidates apply different technologies to satisfy the same requirements or functions. For instance, the PHM system can implement the communication function among different modules within the system boundary via the point-to-point technology (candidate A) or broadcasting technology (candidate B). Such selection requires criteria to evaluate these two candidates.

In some cases, the "preferred architecture" is not the one which delivers the highest performance. For example, a power supply system can be configured with a power supply bus as candidate A, or it can be configured with two power supply buses and an auxiliary power device (e.g. battery) as candidate B, to build the set of system implementation options. In this case, the candidate B has a highly robust configuration with the consideration of redundancy (two power supply bus) and auxiliary power solution (battery) for emergency events, which is able to improve the availability and reliability of a system, in comparison with candidate A. However, the candidate B may have the issues of over-weight and more cost. In the view of that, the preferred architecture selection depends on the trade-off process and the specific constraints in a project. If it is a system that not requires redundancy and auxiliary power, the preferred architecture may be candidate A due to its low-cost and acceptable performance.

a) Task 2.7: Formal system architecture specification

Architecture and design activities require spending several iterations from functional/logical architecture definitions to physical architecture definitions and vice versa until both functional/logical and physical architectures are exhaustive and consistent [38]. Multiple iterations of these activities feed back to the evolving architectural concept as the requirements flow down and the design matures. However, the times of iterations are generally limited due to technical or managerial considerations. The need for further iterations is generally tied to project milestones and reviews. Finally, the technical material in this process is documented, which consists of the functions hierarchy, functional/logical architecture description, physical architecture description, traceability, and analysis evidence, like the initial/updated version of system architecture specification.

2.1.3. Task 3: allocate requirements to architecture elements to form derived requirements

In practice, "system architecture development and the allocation of system requirements to item requirements are tightly coupled and iterative processes, and in each cycle, the identification and understanding of derived requirements increases and the rationale for the allocation of system-level requirements to hardware or software at the item level become clearer" [38]. Derived requirements are requirements that are not explicitly stated in the set of stakeholder requirements, yet are required to satisfy one or more of the stakeholder requirements. They arise from constraints, consideration of issues implied but not explicitly stated in the requirements baseline, factors introduced by the selected architecture and the design. These requirements become the basis for the solution-specified requirements for the system model and are a 'design-to' requirement for the system [45]. In this process, such requirements supplement the system requirements specification to improve the maturity of development life cycle.

2.2. Validation and verification considerations

The process of validation and verification is required to ensure that the architecture definition satisfies the requirements and constraints, by a correct and complete representation of architectural characteristics. Validation is the set of activities ensuring and gaining confidence that a system is able to accomplish its intended use, goals, and objectives (i.e., meet stakeholder requirements or top-level functions) in the intended operational environment [35]. Several methods can support the activity of validation, including traceability, analysis, modeling, test, similarity and engineering review. For example, modeling and simulation used during architecture definition can significantly verify the design items and reduces the risk of failure in satisfying the system mission and performance requirements. Wheatcraf [46] defines that verification refers to the basics (structure) of the item, making sure it meets requirements that drive the creation of the item, standards and best practices (external and internal) on the design, or requirements on the system. General verification methods consist of inspection or review, analysis, modeling, test or demonstration, and service experience. The objectives of validation and verification in the architectures definition process are identified as follows [46]:

a) Confirm that the intended functions have been correctly and completely structured in functional architecture.
b) Examine the behaviors and the transmission of the states of a system.
c) Check the compliance of the defined elements and interfaces.
d) Confirm that the requirements (a group the requirements as a set of functions) have been satisfied.
e) Inspect the consistency during development.

Several papers have discussed the methods of validation and verification for complex systems [47–49]. Furthermore, some literature has discussed the state of the art regarding validation and verification issues in diagnostic, prognostic and health management research [50–52].

3. Application towards PHM system architecture development

In this section, a generic PHM architecture is developed in accordance with the proposed methodology discussed in Section 2.

3.1. Framework

A PHM system involves the specific processes for predicting future behavior and RUL of the monitored system, within the context of the current operating state, future operations and the scheduling of required maintenance actions to maintain systems health [36]. In this paper, a three-layer framework of PHM system is defined to effort aircraft maintenance (e.g. covering systems and components such as the engine(s) and landing gears) services, as shown in Fig. 4. This framework is split into three layers: the onboard layer (aircraft systems, engines, and monitoring sensors), the communication layer (aircraft transmitted system and networks) and the ground layer (airline/manufactures' ground mainframe computing system and PHM system).

3.2. System functions

The system function is the intended behavior of a product according to a set of requirements regardless of implementation in the guidance of ARP4754A [53]. In accordance with the top-level objectives of PHM systems, this paper recommends the top-level functions of the PHM system consist of:

- F1-Data Acquisition (DA)
- F2-Data Processing (DP)

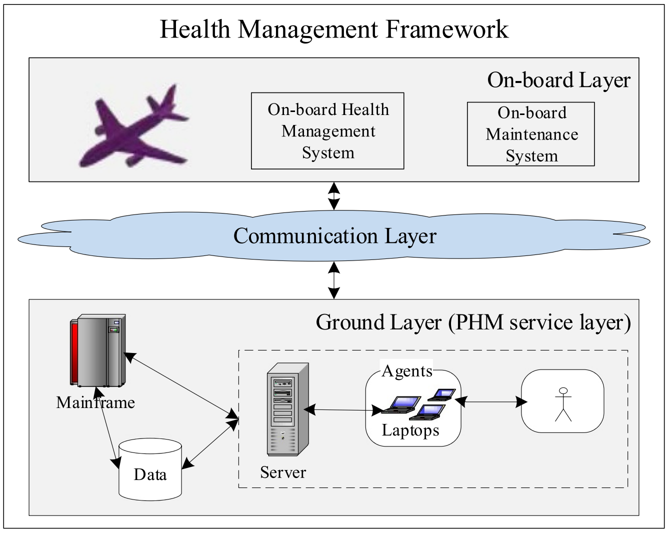

Fig. 4. Conceptual design of the framework.

- F3-Fault Diagnostic Assessment (FDA)
- F4-Prognostic Assessment (PA)
- F5-Health Management (HM)

As shown in Fig. 5, the PHM system has the capability of data acquisition (DA) to collect a significant amount of information from the various in-flight systems (e.g. engine data, sensor data, fault reports, pilot reports) [27]. Once the information is obtained, a data processing (DP) function is able to transmit them to configured functions after necessary manipulations produced on the raw data. On one hand, when the data is transmitted to the fault diagnostic assessment (FDA) function, it has the capability to determine the state of a component or system. This is performed on the basis of fault detection, fault isolation and fault identification by dedicated algorithms [54]. On the other hand, the prognostic assessment (PA) function performs prognostic assessment which includes health state estimation, as well as predicting and determining the useful life of a component/system by modeling the degradation progression in accordance with the operational data [55]. Finally, the health management (HM) function has the capabilities to generate informed and appropriate maintenance advisor via analyzing the assessment information (e.g. state assessment, health assessment, environment and operations).

The top-level functions are systematically decomposed into subfunctions by functions hierarchy diagram which shows all the functions involved in the system in a hierarchical manner, as shown in Fig. 5. Additionally, one hypothesis is that these functions have the characteristic of robust partitioning. The partitioning means that an architectural technique provides the necessary separation and independence of functions or applications to ensure that only the intended coupling occurs. The process of separating, usually with the express purpose of isolating one or more attributes of the software, prevents specific interactions and cross-coupling interference.

3.3. Functional architecture

The health management system of aircraft is composed of onboard systems and ground-based systems, to sustain enhanced information for fault forecasting, troubleshooting, and maintenance history with the help of real-time flight data, so as to decrease scheduled maintenance on the ground and increase the maintenance efficiency.

PHM systems are typically, but not necessarily, defined as being a ground-based health management system (off-board PHM), which is the option pursued here. As a consequence, a generic functional architecture of PHM system is defined in Fig. 6, in accordance with the

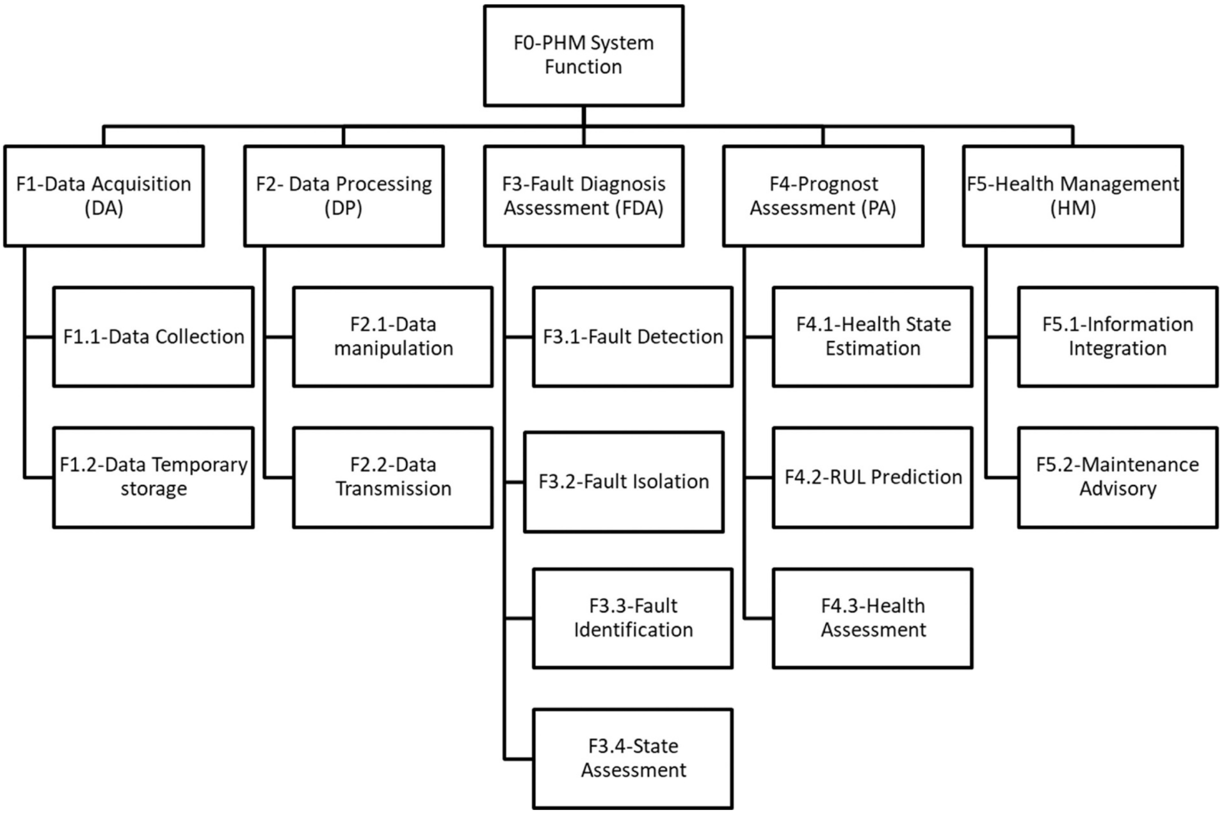

Fig. 5. PHM function hierarchy diagram.

advance research of literatures [26,27,56]. This figure identifies the system boundary and the decomposed functional elements, as well as the internal interfaces among the elements and external interfaces with other systems.

3.3.1. External systems and interfaces

The external systems include in-flight health management system (on-board layer), data sharing network (communication layer), as well as the database model and maintenance management system (ground layer).

a) In-flight health management system

The in-flight health management system is responsible for providing the in-flight data of the aircraft systems/components to ground systems, which primarily include sensors data, condition information, operation data, the various fault reports, maintenance information, historical data, real-time parameters, pilot reports, engine data, etc. As shown in Fig. 6, the in-flight health management systems consist of indicating/recording system, onboard maintenance system (OMS), power plant health management system and data management system to collect flight information, and then all the collected data are transmitted to ground facilities through the aviation data-network system [55]. Furthermore, the flight data management (FDM), also referred to as flight operations quality assurance (FOQA), is the process of collecting and analyzing data form flights to improve safety and efficiency of flight operations, and aircraft design/maintenance [57]. Data recordings are done on a regular basis in order to reveal situations requiring corrective actions before problems occur.

a) G1-Data Sharing Network

The data sharing network provides the communication services between onboard system and ground-based systems, e.g. aircraft communications and reporting system (ACARS). A high-capacity wireless Gatelink system is another way of communication. Airlines are expected increasingly to use wireless datalink systems when they dock at airport gates to downlink aircraft diagnostic and operational data, and simultaneously uplink data to the aircraft's onboard computers, electronic flight bags, and other in-flight systems.

a) G2-Ground Database

The ground database stores the in-flight data and maintenance information data, as presented in Fig. 6. It mainly provide the technical data (e.g. aircraft design data, safety report, manuals, etc.), operation data (e.g. airline operation, monitoring data and sensors data, etc.), maintenance data (maintenance schedule/plans and maintenance history records, etc.) and resources data (spare parts resource, inventory information and manpower resource). More specific, this database collects the operation data from the in-flight system via the data acquisition function, and it acquires the maintenance and resources data from the maintenance management system.

a) G3-Maintenance Management System

When advisories are generated, PHM system will communicate with maintenance services systems to perform the required maintenance actions and services for specific aircraft components or systems. The maintenance services system has the capabilities to update the maintenance schedule, manage inventory and logistics services, and manage maintenance actions.

3.3.2. PHM internal elements and interfaces

The internal elements and interfaces of PHM system, integrated to perform the configured functions, are identified in Fig. 6. The internal elements with different functional characteristics include: data acquisition, data processing, fault diagnostic assessment, prognostic

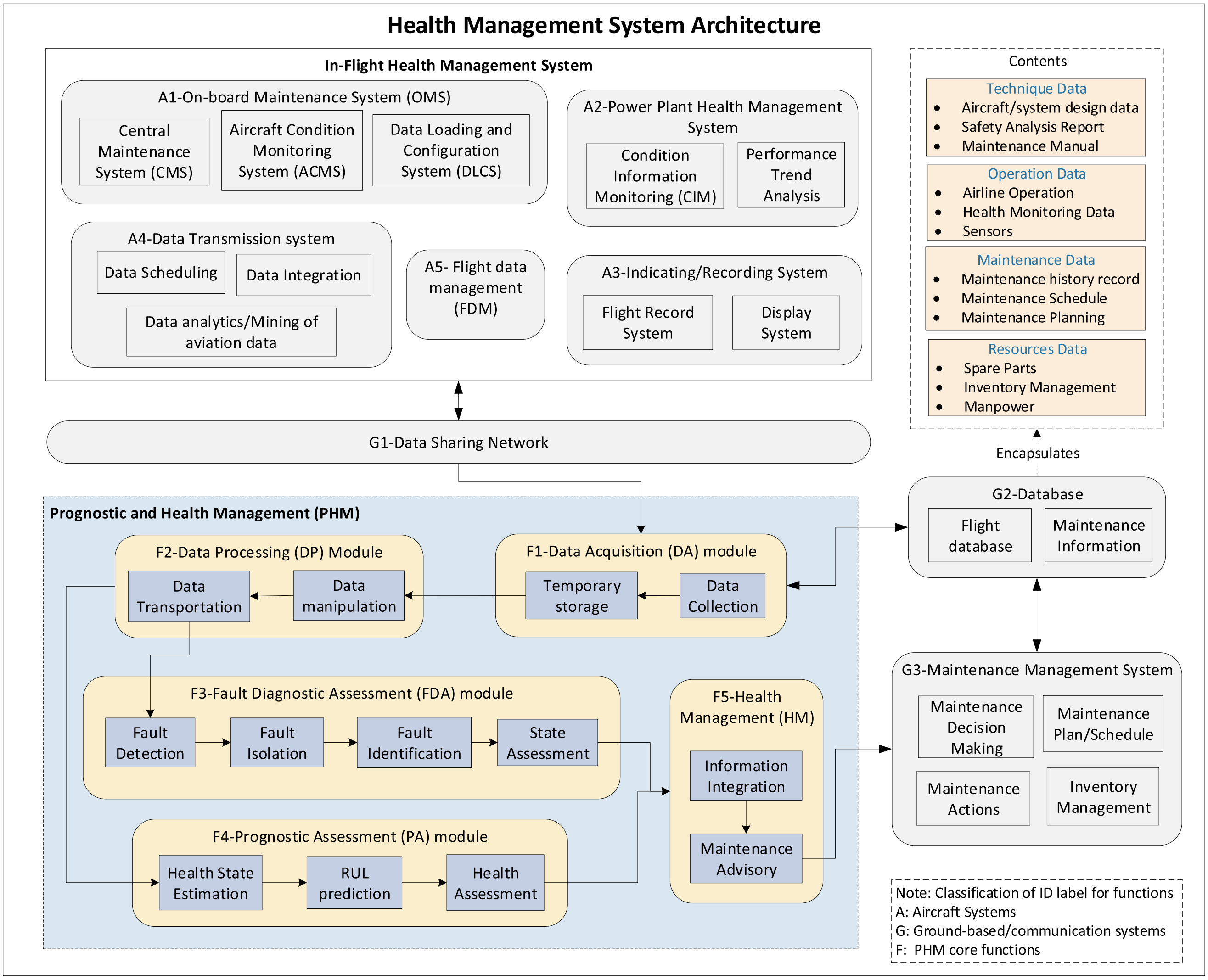

Fig. 6. Functional architecture of PHM system.

assessment and health management [55].

a) F1-Data Acquisition (DA)

The DA module has been generalized to represent the software module that provides the capability to collect the sensors and operational data from the in-flight systems through the data sharing network [58]. Then the collected data will be temporarily stored for further producing by data processing module. Therefore, this functional model has the capabilities of data collection and data temporary storage are identified in Fig. 6.

a) F2-Data Processing (DP)

The DP module is responsible for manipulating the data to a desired form which characterizes specific descriptors (features) of interest in the machine condition monitoring and diagnostic process [58]. This function can be configured with algorithms to perform the signal transformation (e.g., Fast-Fourier Transforms (FFT), and digital filtering), synchronous and nonsynchronous averaging, computations and feature extraction. Afterward, the processed output data will be transmitted to both fault diagnostic assessment module and prognostic assessment module for further analysis.

a) F3-Fault Diagnostic Assessment (FDA)

In Fig. 6, the FDA module implements the functions of fault detecting, fault isolation, and fault identification by software programming configuration. Then, the assessment results (health states) are sent to the health management module for decision making. When appropriate data is available, the state assessments are obtained based on operational context, sensitive to the current operational state or operational environment [58]. Moreover, it also identifies the current operation of the component or system and diagnoses existing fault conditions to determine the state of health and potential failures.

a) F4-Prognostic Assessment (PA)

As a crucial function, the PA module is embedded with a set of prognostic algorithms to perform the functions of health state estimation, RUL prediction and health assessment, as shown in Fig. 6. In this sense, the configured algorithms can be model-based, data-driven or hybrid prognostic approaches for specific system/components (e.g. engine, landing gear, bearing, etc.). The PA function is able to determine the current health state on account of analyzing the features extracting from the selected sensors data. The objective of this function is to determine the current health state and estimate the further status, in order to predict the remaining useful life by modeling failure progression on the basis of the extracted features from the historical data. Lastly, it will publish the health assessment report to HM module for maintenance advisory generation.

a) F5-Health Management (HM)

The primary function of HM module is able to integrate the information from FDA and PA functions and consolidate with the constraints (safety, environmental, budgetary, etc.) to provide recommended maintenance advisories to an external maintenance management system. At this point, the PHM system has completed its mission. Afterward, the maintenance management system, outside the PHM boundary, is responsible for making the maintenance decisions with the consideration of resources (inventory, parts, and human resources) and maintenance plan/schedule, and managing the maintenance services.

3.4. *Physical architecture*

The physical architecture concretizes physical elements that can sustain functional, behavioral and temporal features along with the expected properties of the system deduced from non-functional system requirements (e.g. constraints, replacements, configuration, and/or continued product support) [43].

A set of system requirements and a functional architecture can drive more than one physical architecture candidate depending on the different technologies available for physical implementation. The preferred candidate is then selected based on the constraints of a specific project. For simplicity, this paper recommends one physical architecture candidate compliant with the functional architecture for a PHM system in Fig. 7. In Fig. 7, this generic physical architecture of ground-based PHM system incorporates three main modules for implementation, which are the cabinet, auxiliary power module and integrated computing module. In reality, these physical modules can be Commercial off-the-shelf (COTS) products satisfying the needs of a specific project. The integrated computing module is incorporated with hardware elements and software elements. The hardware elements (e.g. operating equipment and embedded sensors), can provide the computing capability, resources and operating environment. The embedded sensors cooperate with in-flight sensors, which are capturing critical data for processing and feeding user interfaces that present key metrics and intelligence to an operator/user so they are appropriately informed of changing health conditions. Additionally, the software elements, including the main computing functions and support functions as the core, enable to implement the required functions and provide the related services. For example, the software element can be configured with a set of diagnostic or/and prognostic algorithms to perform fault isolation, identification and RUL prediction for different aircraft components and systems. The cabinet is an enclosure with fitted, fixed or removable modular slots integrated with bus and power supply, which provides the data bus to connect with database and other ground mainframe systems.

3.5. *Requirements derivation and allocation*

In practice, system architecture development and the allocation of system requirements to item requirements are tightly coupled iterative processes. With each cycle, the identification and understanding of derived requirements increases and with it the rationale increases for the allocation of system requirements to functional elements and physical elements (hardware or software).

Upon definition of PHM generic architecture, the specific requirements or a group of requirements (function) should be allocated to the related architecture elements. Meanwhile, a set of derived requirements may be generated as a result of architectural design decisions, which represent the factors of elements, data flows, interfaces, behaviors and so on. In this case, the physical architecture generates some derived requirements in accordance with the implementation selection and design constraints. The following are some examples of possible derived requirements in this application:

- PHM-DR-1: The PHM system should provide at least 64 GB memory for temporary storage.
- PHM-DR-2: The PHM system should provide a redundant power supply.
- PHM-DR-3: The PHM system should be able to accommodate multiple prognostic algorithms.
- PHM-DR-4: The PHM system should be configured with a 64-bit data bus.
- PHM-DR-5: The PHM system should be able to call for a specific algorithm as defined in configuration files.

3.6. *Architecture validation and verification*

As aforementioned, validation is the process of ensuring the architecture is correct and complete, and ensuring compliance with requirements or stakeholders expectations; besides, verification ensures that an item within architecture complies with all of its design options. In this research, when we obtain the PHM architecture, the validation and verification activities can ensure the confidence that the defined architecture is able to accomplish the intended functions of PHM system, and compare that the architecture against the required

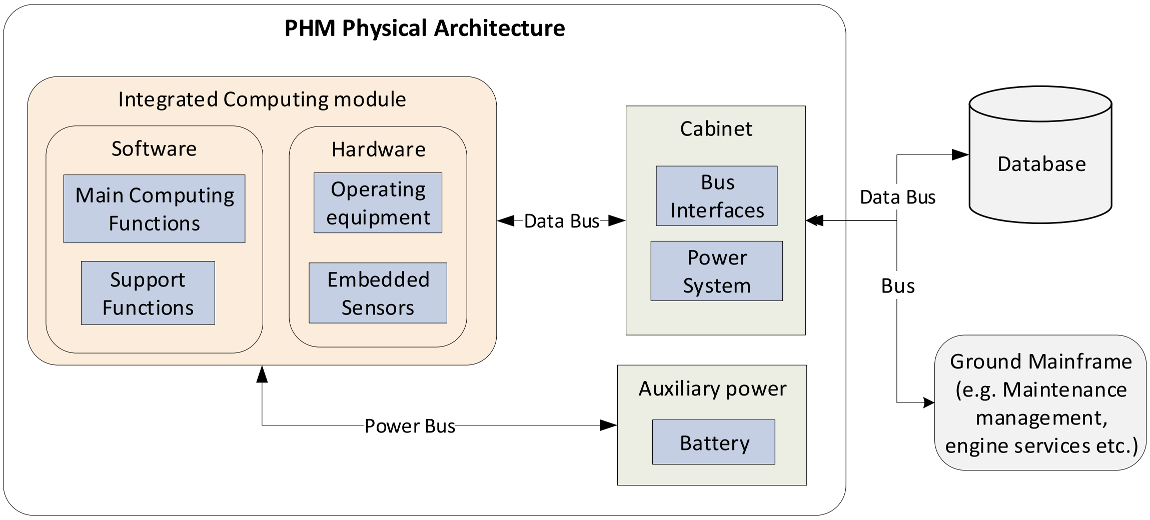

Fig. 7. Physical architecture of ground-based PHM system.

Table 1
Verification matrix of SysML modeling method.

Means	Items	Activity description	Objective	
4.1 Functional Structure Modeling	Block Definition Diagram (BDD)	Functional elements	Identify and check compliance of the functional elements with functions hierarchy.	Verification
	Internal Block Diagram (IBD)	Functional interfaces	Identify the internal connections between functional elements and the details of how elements are connected with each other.	Verification
4.2 Logical Behavior Modeling	4.2.1 State Diagram	State transition	Describes the discrete states of elements and the transition from one state to another, which is used to capture the operational scenarios of a system.	Verification
	4.2.2 Activity diagram	Activity	Identify and check the system activities, data flow, and control flow in sequence with the logical relationships.	Verification
4.3 Physical Structure Modeling	Block Definition Diagram (BDD)	Physical elements	Identify the physical elements of a system and check the consistency of requirements and functions.	Verification
		Physical interfaces	Check the correctness and completeness of the interfaces between the physical elements, and examine the consistency with system behaviors.	Verification

characteristics. To achieve these objectives in Section 2.2, the specific validation and verification method used in this case are presented in details through Tables 1 and 2.

Furthermore, Section 2.2 has discussed a set of validation and verification means respectively, from SE perspective. For example, analysis and modeling methods can be applied to verify the PHM architecture conducted in this section. The modeling of complex systems typically consists of a combination of computation analysis and tests; however, modeling deterministic systems behavior may also be entirely computational, and is usually able to examine the behaviors and states transmissions, as illustrated in Table 1. Moreover, an analysis method provides evidence of compliance by performing a detailed examination (e.g., functionality, performance, interfaces) of a system or element, as present in Table 2. Similarly, the methods of traceability and analysis can be used for validation activity, as the described in Table 2, with the purpose of checking the compliance of defined elements and interfaces, as well as the consistency.

Firstly, the case study utilizes the SysML modeling method to check, analyze and exam the design elements and interfaces of the architecture from functional, logical and physical perspective views [59]. SysML is a modeling language for engineering systems which is able to build models for system specification, design, analysis, validation, and verification as expressed in the papers [60–63]. SysML can represent systems, components, and other entities as follows:

- Structural composition, interconnection, and classification;
- Function-based, message-based, and state-based behavior;
- Constraints on the physical and performance properties;
- Allocations between behavior, structure, and constraints (e.g., functions allocated to components);
- Requirements and their relationship to other requirements, design elements, and test cases.

In this paper, Table 1 shows the matrix of verification items by various modeling diagrams. These verification means are primarily applied in the case study of PHM architecture modeling, as further discussed in Section 4.

Furthermore, a set of various analysis means are used to validate and verify the generic PHM architecture for the items of functions, interfaces, traceability, and compliance with Table 2 [64]. In reality, one item is always validated and verified by more than one means in order to improve the confidence of validation and verification results. Accordingly, the means of functions analysis in Table 2 is a parallel activity with respect to structure modeling defined in Table 1, to validate and verify the same items (functional element and interface) in architecture definition. Similarly, the means of interfaces analysis is conducted in parallel with structure modeling to check the item of system interfaces. To conclude, these validation and verification means are applied in the case study of PHM architecture analysis, as presented in Section 5.

4. Case study 1: PHM architecture SysML modeling

In this section, a case study is conducted to validate and verify the design of the PHM system architecture based on Table 1. This case study establishes a modeling project for PHM system to model the PHM functional architecture, logical behavior and physical architecture using different diagrams via the tool of Enterprise Architecture [59].

4.1. Functional structure modeling

The block definition diagram (BDD) is a black-box structure of the system, with the connections between components and external interfaces, and the interfaces present a whole part or composition, or communication relationship between the blocks [65]. The functional elements related to PHM system are modeled in SysML and presented

Table 2
Validation and verification matrix including analysis methods.

Means	Items	Activity description	Objective
5.1 Functions Analysis	Functions decomposition, elements and interfaces	Check the functional elements in consequence, and the functions are organized and depicted by the order of execution.	Verification
5.2 Interface Analysis	Interfaces	Check the interface/interactions between each functional element and the external interfaces with other systems.	Verification
5.3 Traceability Analysis	Sources of the design items	Traceability matrix from functions/requirements to functional elements and physical elements ensure that the architecture is compliant with the high-level requirements.	Validation
5.4 Compliance Analysis	5.4.1 Compliance with ISO-13374	Check the compliance between the PHM architecture and the standards of ISO-13374 to demonstrate its interoperability and compatibility with various systems.	Validation
	Compliance with IEEE Std 1856	Check the compliance with IEEE standard 1856 to demonstrate the effective applicability of the PHM architecture.	Validation

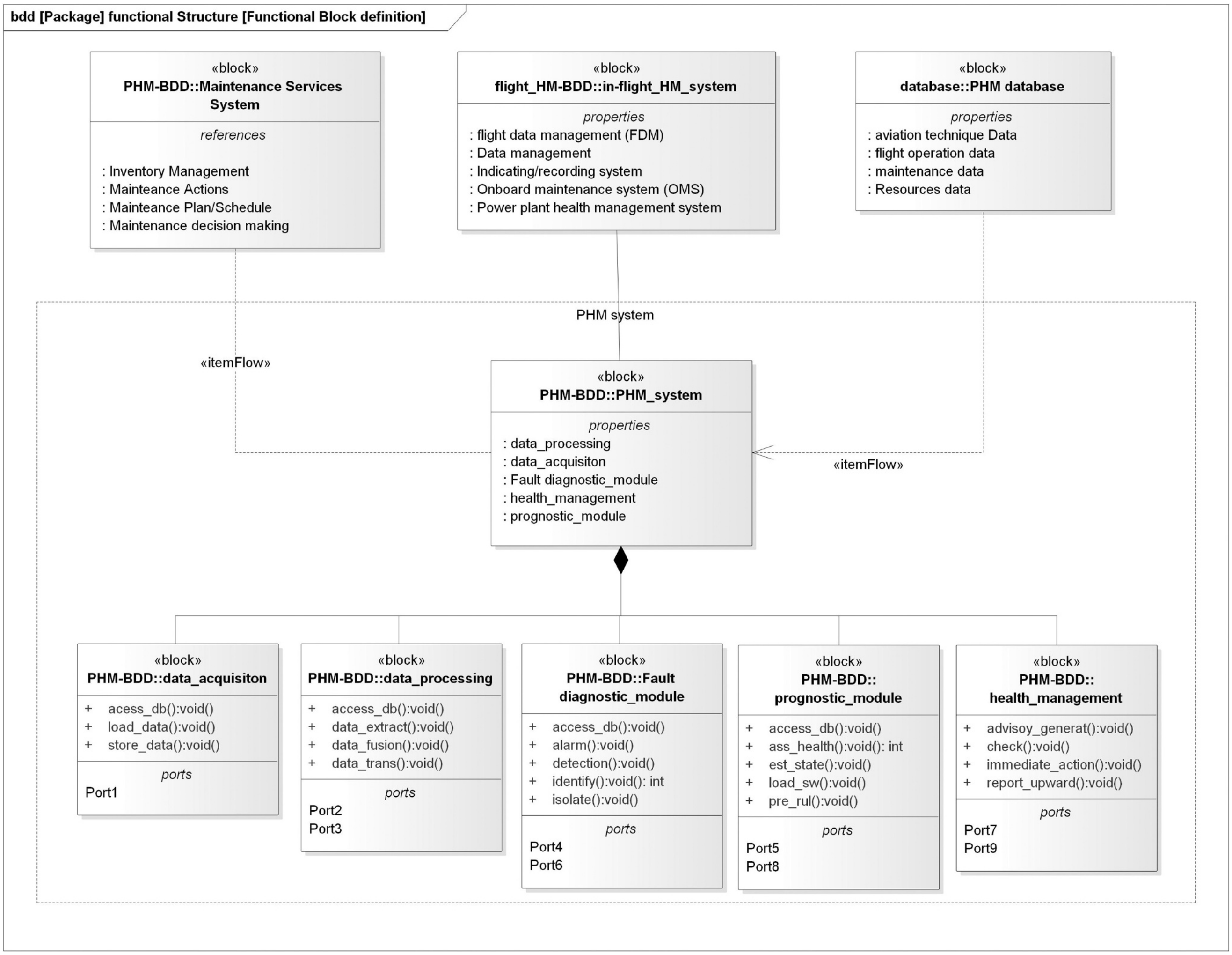

Fig. 8. Block definition diagram of PHM system.

through the BDD diagram as given in Fig. 8. This modeling diagram is compliant with the elements addressed in Fig. 6. The PHM system is connected to the external systems, such as in-flight health management systems and the ground database [59]. It also connects with the related maintenance services system in order to perform the required maintenance services and actions. Further, it also identifies the partitioning of functional blocks within the PHM system boundary, including the blocks of data acquisition, data processing, fault diagnostic assessment, prognostic assessment and health management, as illustrated in Fig. 8. Additionally, each block has the functionality to configure the specific attributes, operations and ports information, which can be used for testing and simulation in further research.

The Internal Block diagram (IBD) can be used to define the internal connections between parts, and the details of how parts wired with each other. The IBD modeling diagram of PHM system, which provides more details regarding the specific nature of the relationship between blocks, has been addressed in previous research [55].

4.2. Logical behavior modeling

System logic modeling is able to describe the logical relationships and data-flows among the partitioning elements within the system boundary. This case study constructs the state machine diagrams and activity diagrams for logical behavior modeling in the following sub-

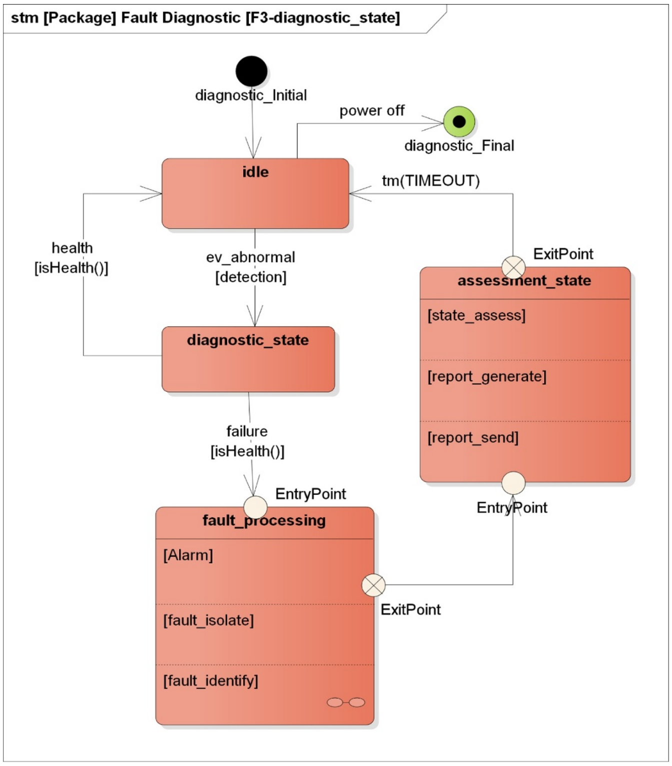

Fig. 9. State machine diagram of FDA function.

sections, which are all compliant with the structure in Fig. 6.

4.2.1. State diagrams

A state machine diagram describes the discrete states of a block and the transition from one state to another, which presents a condition of a block. The transition between these states may be triggered by the receipt of a configured signal or behavior, such as a time-based event or customized event [65]. In this paper, the state diagrams for the functions of fault diagnostic assessment (FDA), prognostic assessment (PA), and health management (HM) are modelled as examples.

a) F3-Fault Diagnosis Assessment (FDA)

As aforementioned, the fault diagnostic assessment function is responsible for fault detection and isolation. Therefore, this function is configured with the states: Ideal, diagnostic, fault processing and assessment, as shown in Fig. 9. Initially, it is in the idle state, waiting for events and preparing to receive corresponding events to call the other operations. When it receives abnormal events, the diagnostic state is activated to implement the diagnostic procedures for detecting the faults [26]. Once any fault is detected, the fault processing state is activated, in which the system triggers the alarm, isolates and identifies the specific faults. It subsequently transmits to the assessment state to assess the operational status and send an assessment report to the health management function.

a) F4-Prognostic Assessment (PA)

Similarly, the prognostic function is responsible for prediction of future health status and estimation of the RUL for the target system. Thus, this function is configured with the states: idle, prognostic, rul_prediction and assessment, as illustrated in Fig. 10. When the module initializes, it stays in the idle state, and then it will transit to prognostic state according to the time cycle interval. In the prognostic state, the prognostic procedure runs periodically to predict degradation trends via the designated algorithms. Subsequently, it performs procedures for estimating the RUL in the rul_prediction state. Ultimately, the assessment state will analyze the health status of the target system and forward the assessment report to health management function for maintenance advisory making [26].

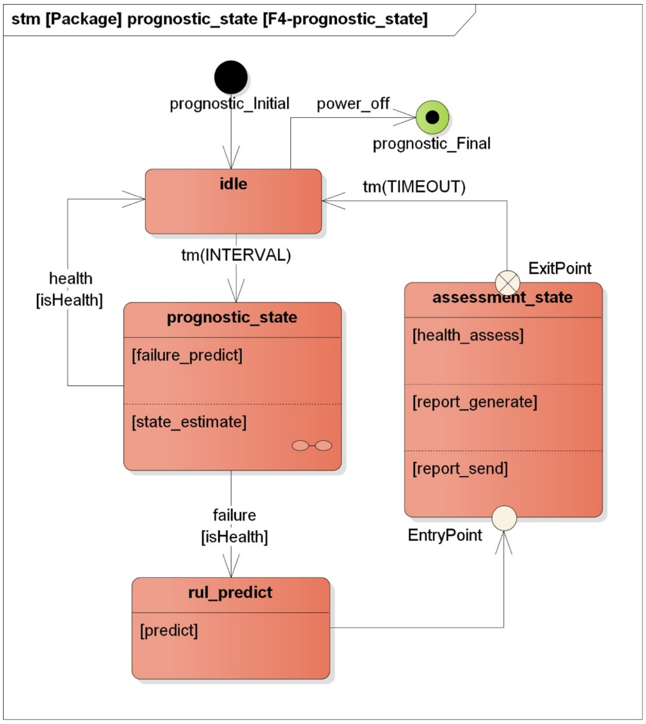

Fig. 10. State machine diagram of PA function.

a) F5-Health Management (HM)

The health management module has the capability to generate the maintenance advisories and manage the maintenance services, so it is configured with the states: idle, Gen_advisory, maintenance, as addressed in Fig. 11. This function starts as the idle state when power is switched on. After initialization, it is in the idle state waiting for reception of the configured events of assessment reports, to activate the maintenance advisory state. Subsequently, it performs the procedures to analyze the health status reports and generate maintenance advisories. Once the maintenance advisories are generated, the maintenance state is activated to decide the transmission to external health services system for future maintenance decision making and required maintenance actions.

4.2.2. Activity diagrams

An activity diagram transforms a set of inputs to outputs through a controlled sequence of actions. It means that the activity diagram describes how these activities perform the defined functions on account of certain sequences and logic decisions, which reflect the operational procedures to provide the services [65].

As a core element, Fig. 12 presents the primary functionalities and activities flow of PHM functions. It identifies the fundamental control flow, decision nodes and related events and actions among the operational activities in a logical sequence, which also contributes to understanding the functions and operation process of PHM system [26]. For example, as shown in Fig. 12, the fault diagnostic assessment function performs the activities of fault isolating and identifying, when any faults are detected. Once a catastrophic fault is detected, it immediately provides an alarm for an emergency event. In another example, the health management function can integrate all assessment information from both diagnostic and prognostics functions. Then, it

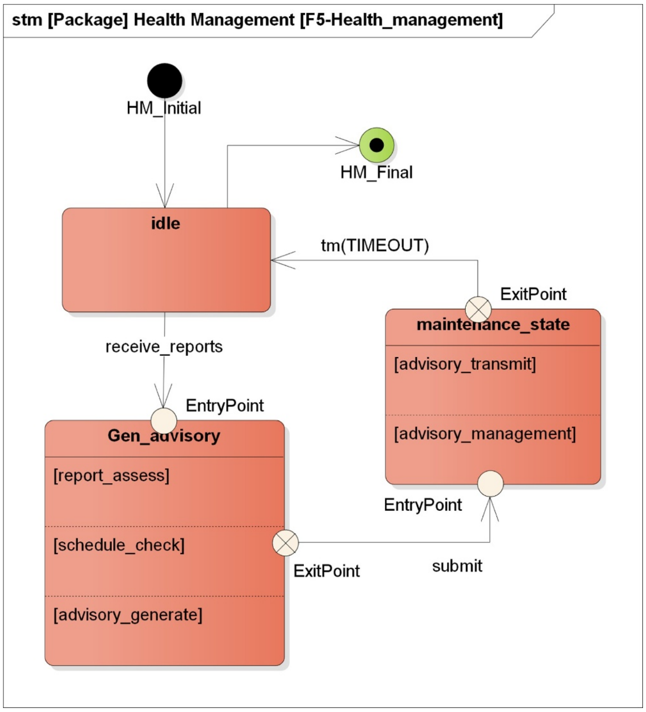

Fig. 11. State machine diagram of HM function.

integrates all the health information and provides a maintenance advisory. When the operation and maintenance advisory is generated, it will communicate with the external system which is responsible to support subsequent decision making and/or execution.

4.3. Physical structure modeling

Fig. 13 is the BDD modeling diagram of the PHM physical architecture defined in Fig. 7. This diagram identifies the decomposition of PHM system with the elements of a cabinet, auxiliary power modules, and integration computing modules. More specifically, the integrated computing module has the capability to load and install the configured software code to perform a set of algorithms through the devices of CPU and memory. Besides the battery module auxiliary power devices can provide the power to the computing module when the power supply is shut down in emergency cases. A cabinet is equipped with fixed power supply and data buses, as well as multiple computer racks for mounting integrated modular based on a configuration in a specific project. Furthermore, the relationship of how to implement the defined functions based on operating activities, identified in Fig. 12, through the defined physical elements and interfaces are present in Table 4.

5. Case study 2: PHM architecture analysis

Based on the matrix in Table 2, this section describes a case study to validate and verify the design of a generic PHM system architecture through various analysis methods, such as functions analysis, interfaces analysis, traceability analysis and compliance analysis.

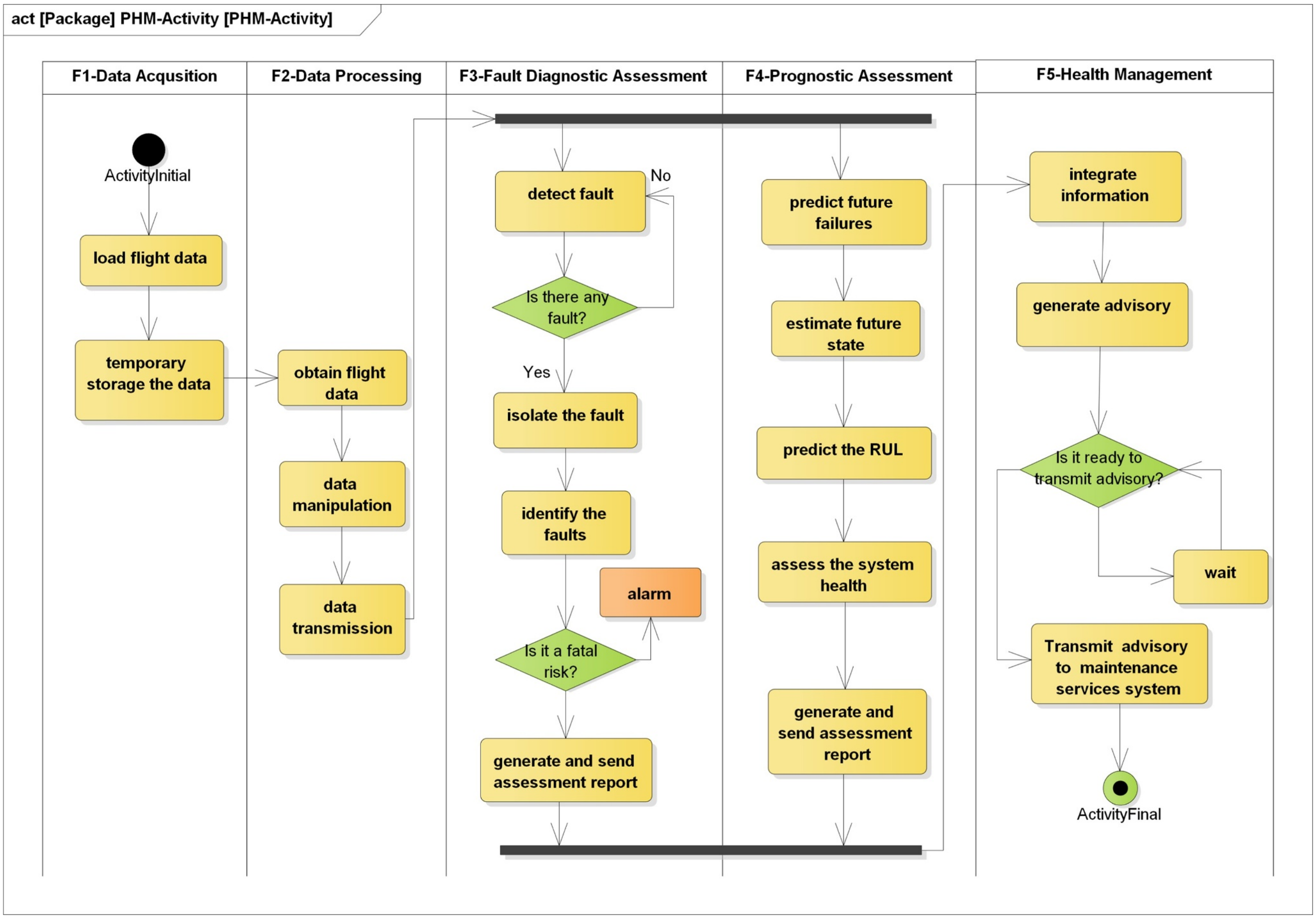

Fig. 12. Activity diagram of PHM system.

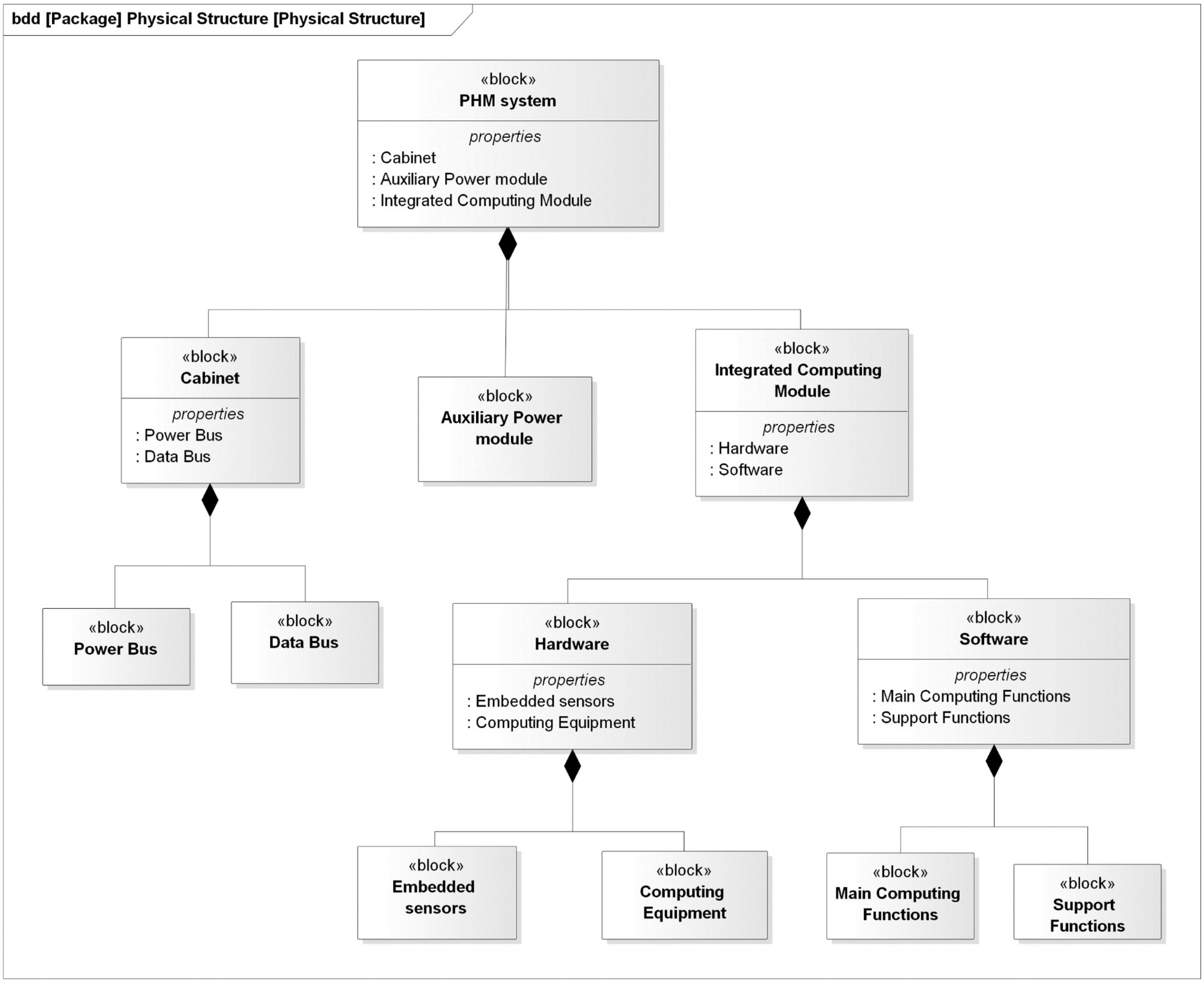

Fig. 13. PHM physical architecture modeling.

5.1. Functions analysis

Functional analysis is utilized to ensure that all functional elements of the system are described, recognized and defined. The functional flow block diagram (FFBD) is a time-sequenced and step-by-step diagram of the system's functional flow, with the detailed, operational and support sequences for the system. In other words, FFBD enables showing the sequential relationship of all functions that should be accomplished and identifies functional interfaces. In an FFDB diagram, the functions are arranged in a logical sequence so that any specified operational use of the system can be traced in an end-to-end path. For example, some functions may be performed in parallel or alternate paths.

Firstly, the top-level functions flow block diagram is defined in Fig. 14. In this diagram, the top-level functions are organized and depicted by their logical order of execution, and each of them is represented with the logical relationship (e.g. Logic symbols represent, the sequential or parallel execution) to the execution and completion of other functions.

Furthermore, Fig. 15 analyzes and identifies the low-level functional flow block diagrams for PHM system, which is compliant with the top-level functions flow in Fig. 14 and the decomposition of PHM functions

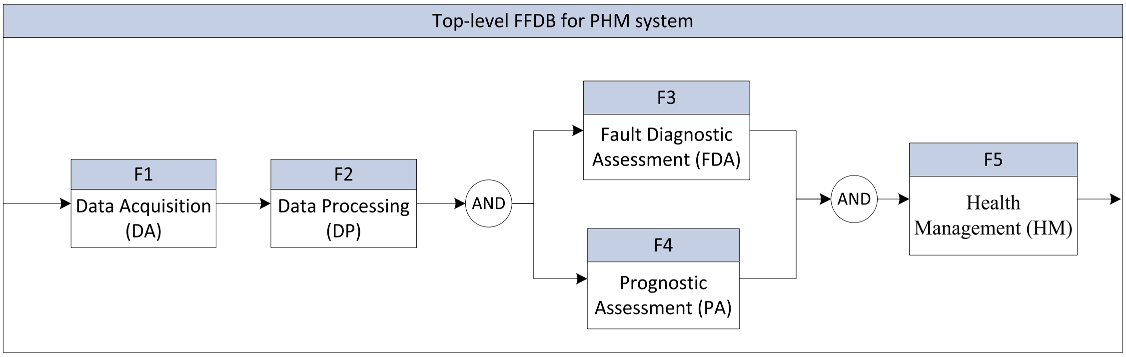

Fig. 14. FFBD diagram of top-level functions for PHM system.

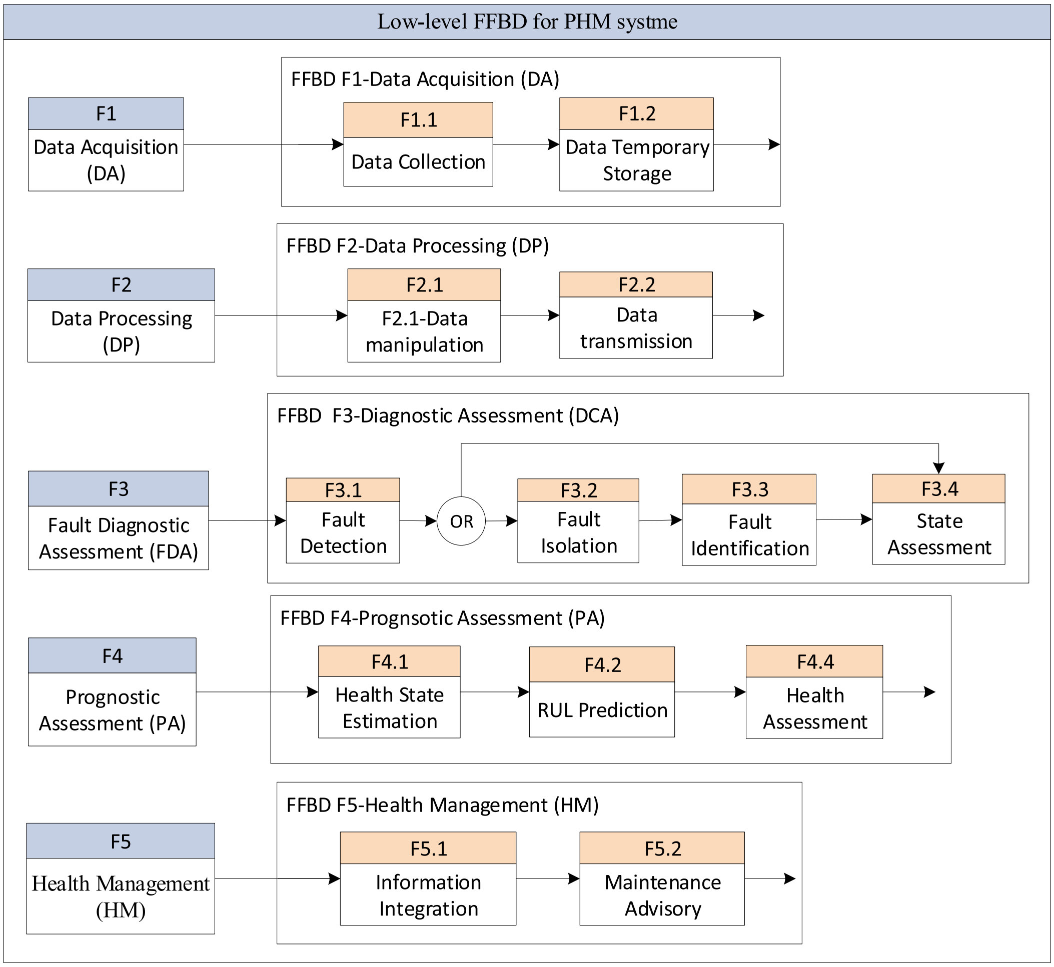

Fig. 15. FFBD diagrams of low-level functions for PHM system.

Table 3
Interface mapping for PHM system.

		A1	A2	A3	A4	A5	G1	G2	G3	F1	F2	F3	F4	F5
On-board Maintenance System (OMS)	A1	A1	X	X	X	X								
Power plant health management system	A2	X	A2	X	X	X								
Indication/Recording system	A3	X	X	A3	X	X								
Aircraft data transmission	A4	X	X	X	A4	X	X							
Fight data management (FDM)	A5	X	X	X	X	A5								
Data sharing network	G1				X		G1							
Database	G2							G2	X	X				
Maintenance Management System	G3							X	G3					X
Data Acquisition (DA)	F1						X	X		F1				
Data Processing (DP)	F2									X	F2			
Fault Diagnostic Assessment (FDA)	F3										X	F3		
Prognostic Assessment (PA)	F4										X		F4	
Health Management (HM)	F5											X	X	F5

in Fig. 5. Differently, this diagram emphasizes the end to end functional flow within the PHM system. For instance, the DA function firstly collects and stores the in-flight information data. Then the collected data is transmitted to both fault diagnostic and prognostic functions after the necessary procedures via the DP function. The FDA function detects a fault, and then it is able to isolate and identify when there is any abnormality. It then assesses the health status of the target system and consolidates a diagnostic assessment report to submit to the health management function. Meanwhile, the PA function is responsible for sequence predicting the failure, and predicting RUL for the monitored system. Besides, this function also provides the health assessment report to health management function. Finally, the HM function is responsible for generating maintenance advisory according to the integrated health state information.

5.2. Interface analysis

During system design, it needs to be defined how the system is required to interact or to exchange material, energy, or information with external systems (external interface), or how system elements within the system, including human elements, interact with each other (internal interface) [38]. Generally, the interface definition is performed along with the architecture definition process and is refined during architecture iteration activities. This includes both internal interfaces

Table 4
Traceability analysis matrix.

Function	Requirements	Functional element	Logical analysis	Physical element
F1-DA	Related functional and performance requirements of data acquisition.	Data Acquisition module	Activity diagram to analysis	Integrated computing module
F2-DP	Related functional and performance requirements of data processing.	Data Processing Module	Activity diagram to analysis	Integrated computing module
F3-FDA	Related functional and performance requirements of diagnostic.	Fault Diagnostic Module	Activity diagram and state machine diagram to analysis	Integrated computing module
F4-PA	Related functional and performance requirements of prognostic.	Prognostic module	Activity diagram and state machine diagram to analysis	Integrated computing module
F5-HM	Related functional and performance requirements of health management.	Health Management Module	Activity diagram and state machine diagram to analysis	Integrated computing module

between system elements (e.g. PA function connects to HM function within the boundary of PHM system), and the external interface with other systems (e.g. DA function interfaces to G1 function, which is external to the boundary of the PHM system). In practice, the system engineers team is responsible for system interfaces definition, with the support from technical specialists.

The interface analysis contributes to the activities of integration, validation, and verification during the development life-cycle for a complex system. Particularly, the activity of interface analysis enables checking the conformance of elements interfaces, and systematic coverage of relations and interfaces. Besides, interface analysis can perform semantic analysis, checking the description of the functionality of each element, and contracts among specifications; and perform behavioral analysis based on the behavior specification, including the activities of checking behavioral conformance and equivalence relations [38].

In this case, the internal and external interfaces among elements of the PHM system are identified in Table 3. It analyzes the interfaces of the PHM system using a three-layer framework incorporating the onboard layer, the communication layer, and the ground layer, and it also identifies the interactions between the elements within each layer. In terms of physical connection, communication among different modular parts is based on the data bus or power supply bus. Moreover, this interface mapping matrix visualizes the internal and external interfaces among the elements, which provide assistance to check the interfaces identified in Fig. 6 in a consistent manner. Conversely, if they are not in conformance, it is also the result of verification.

5.3. Traceability analysis

As discussed in Section 2.2, traceability is one of several available validation methods. Traceability is the recorded relationship established between two or more elements of the development process. For example, traceability can consider the bidirectional traceability of the architecture (e.g. functional elements, internal interfaces, and boundary) characteristics, and the design characteristics (partitioning functions and communication among functions). Therefore, the traceability method is able to ensure the consistency of design items during development.

In this case study, the development traceability starts from requirements (functions) to functional architecture (functional elements, interaction), logical architecture (consequence behavior), and then allocated to physical architecture (physical elements and interfaces), which are summarized in Table 4. This table sufficiently demonstrates that a lower design element or item satisfies a higher level requirement/ stakeholders' expectation (functions) with regards to completeness. In the other words, it is used to make sure the sources of design elements at each development level.

5.4. Compliance analysis

In general, compliance means has the characteristics of conforming to a rule, e.g. a specification, policy, standard or law, which is used to ensure the design quality of a complex system. This case study analyses the compliance between PHM architecture proposed in this paper, and the standards of ISO-13374 and IEEE standard 1856, according to Table 2. The analysis results are able to demonstrate the interoperability, compatibility, and applicability characteristic of PHM system.

5.4.1. OSA-CBM (ISO-13374)

The standard of ISO-13374-1 establishes the general guidelines for data processing, communication, and presentation of machine condition monitoring and diagnostic information [31]. This standard defines an open framework of PHM system, which consists the basic functional blocks organized in layers, including data acquisition (DA), data manipulation (DM), state detection (SD), health assessment (HA), prognostic assessment (PA), and advisory generation (AG), as list in Table 5.

Table 5
Compliance matrix with OSA-CBM [55].

OSA-CBM Functional Blocks	PHM functions				
	F1-Data Acquisition	F2-Data Processing	F3-Fault Diagnostic Assessment	F4-Prongostic Assessment	F5-Health Management
Data Acquisition (DA)	X				
Data Manipulation (DM)		X			
State Detection (SD)			X		
Health Assessment (HA)			X		
Prognostic Assessment (PA)				X	
Advisory Generation (AG)					X

Table 6
Compliance with IEEE Standard 1856 [34].

IEEE Std 1856	Proposed Generic Architecture	
1) Acquisition of object system data (e.g., by means of sensors),	F1-Data Acquisition	
2) Data management, and	F2-Data Processing	
3) Data processing algorithms and/or processes for:	Diagnostics, health state estimation, and prognostics	F3-Fault Diagnostic Assessment F4-Prognostic Assessment
	Health management	F5-Health Management

Standards provide designers/engineers with a basis for mutual understanding and are used as tools to facilitate communication, measurement, commerce, and manufacturing. If the proposed architecture is compliant with the standard, it provides confidence and credibility of the PHM design, and is likely more easily accepted by engineers, practitioners and/or end users.

The proposed generic PHM architecture is defined in Fig. 6 in this paper. This architecture is compliant with the open framework of the PHM system defined in standard ISO-13374-1 (2003), and the results are illustrated in Table 5. Regarding the evaluation, the compliance matrix, in Table 5, ensures the credibility and compatibility of this architecture; what is more, it demonstrates the feasibility of integration and interoperability of the proposed PHM architecture with various other systems.

5.4.2. IEEE standard 1856

The IEEE standard 1856 describes the information for the implementation of PHM for electronic systems [34]. It provides to manufacturers and end users for planning the appropriate prognostics and health management techniques to implement and the associated life cycle operations for the system of interest.

The related requirements of PHM framework in IEEE Standard 1856 are defined as follows [34]:

a) A prognostics and health management system shall consist of subsystems and components with capabilities including:
  1) Acquisition of object system data (e.g., by means of sensors),
  2) Data management, and
  3) Data processing algorithms and/or processes for:
    - Diagnostics, health state estimation, and prognostics
    - Health management

Due to the scope of this paper, this case study analyzes the compliance between the items of PHM framework in IEEE Standard 1856 and the proposed generic PHM architecture, as shown in Table 6. This standard [34], provides a standard framework that assists practitioners in the development of business cases and the selection of related techniques (e.g. approaches, methodologies, algorithms, procedures, etc.) for implementing PHM systems. Accordingly, the proposed PHM architecture has the characteristics of implementation and applicability.

6. Conclusions and future work

This paper proposes a systematic methodology for PHM architecture definition, leveraging the concept of “RFLP” from system engineering and improving upon existing methodologies by providing a more complete and consistent representation of PHM architecture development from start to finish.

A detailed description of a generic PHM architecture is presented in a set of architecture views in accordance with partitioning elements and physical modularity; hence it has the architectural characteristics of functional/physical dimensions, modularity, and robustness. The robust partitioning system allows partitions with different criticality levels to execute in the same module without affecting one another spatially or temporally. The modularity in the design contributes to system reusability and mitigating the risk of duplicate work. These architecture views of a generic PHM system, as defined in Figs. 6 and 7, could be used as a practice case as to how to apply the proposed methodology.

A case study is conducted to verify and validate the proposed generic PHM architecture to ensure correctness and completeness. SysML modeling and various analysis means are employed to this end. More specifically, the SysML modeling diagrams show the start of the design elements, interfaces and the applied relevant techniques, such as the diagrams in Figs. 8 and 13. In addition, the operating functions and sequences of the PHM system are identified on the basis of the defined functional/physical elements (Figs. 9 to 12). To sum up, this case study confirms that the intended functions have been correctly and completely structured in functional architecture; examine the behaviors and the states transmission of a system, and confirms that the set of functions have been satisfied. Another case study analyses features of PHM architecture, expressing that the generic PHM architecture has the characteristics of interoperability, compatibility, and applicability allowing integration with a variety of systems through the means of interface, traceability and compliance analysis. Practical and important issues including functional flow, interfaces, implementations and standards are also covered as part of the required tasks. It also shows how a relatively small set of generic standardized interfaces can provide this interoperability and applicability. This case study ensures the compliance of the defined elements and interfaces, and the consistency of design elements during development.

In conclusion, this research contributes towards a systematic PHM system design and development methodology, including as its main elements:

- Providing guidance toward developing a PHM system, in particular the specification of a PHM architecture (and associated verification and validation thereof) in practice;
- Providing a generic functional and physical architecture of a PHM system using system engineering principles;
- Providing a reusable and practical approach;
- Addressing compliance, consistency, interoperability, and applicability of the proposed architecture.

Future studies on the current topic are concentrated on enhancing a comprehensive understanding of design methodology for a complex system involving requirements, architectures, and design solution and validation/verification items. From the current research, the following items are identified to be vital for contributing to an efficient PHM design:

- Apply and validate the proposed systematic design methodology for PHM architecture development towards industrial case studies;
- Improve the maturity of prognostic techniques in terms of robustness, reliability, and applicability;
- Select the appropriate prognostic approaches based on requirements and project constraints.

Acknowledgment

This work was supported by China Scholarship Council (CSC) [grant number 201609110122].

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.ress.2019.106598.

References

[1] Vogl GW, Weiss B a, Donmez MA. Standards for Prognostics and Health Management (PHM) techniques within manufacturing operations. Annu Conf Progn Heal Manag Soc 2014;2013:1–13.
[2] Brahimi M, Medjaher K, Leouatni M, Zerhouni N. Prognostics and Health Management for an overhead contact line system - a review. Int J Progn Heal Manage 2017:1–16.
[3] de Jonge B, Teunter R, Tinga T. The influence of practical factors on the benefits of condition-based maintenance over time-based maintenance. Reliab Eng Syst Saf 2017;158:21–30. https://doi.org/10.1016/j.ress.2016.10.002.
[4] De Jonge B, Klingenberg W, Teunter R, Tinga T. Optimum maintenance strategy under uncertainty in the lifetime distribution. Reliab Eng Syst Saf 2015;133:59–67. https://doi.org/10.1016/j.ress.2014.09.013.
[5] Do P, Voisin A, Levrat E, Iung B. A proactive condition-based maintenance strategy with both perfect and imperfect maintenance actions. Reliab Eng Syst Saf 2015;133:22–32. https://doi.org/10.1016/j.ress.2014.08.011.
[6] Yoon JT, Youn BD, Yoo M, Kim Y, Kim S. Life-cycle maintenance cost analysis framework considering time-dependent false and missed alarms for fault diagnosis. Reliab Eng Syst Saf 2019;1:1. https://doi.org/10.1016/j.ress.2018.06.006.
[7] Alaswad S, Xiang Y. A review on condition-based maintenance optimization models for stochastically deteriorating system. Reliab Eng Syst Saf 2017;157:54–63. https://doi.org/10.1016/j.ress.2016.08.009.
[8] Liu J, Zio E. System dynamic reliability assessment and failure prognostics. Reliab Eng Syst Saf 2017;160:21–36. https://doi.org/10.1016/j.ress.2016.12.003.
[9] Kim H, Kim JT, Heo G. Failure rate updates using condition-based prognostics in probabilistic safety assessments. Reliab Eng Syst Saf 2018;175:225–33. https://doi.org/10.1016/j.ress.2018.03.022.
[10] Pecht M, Kumar S. Data analysis approach for system reliability, diagnostics and prognostics. Hawaii, USA: Pan pacific Microelectron. Symp.; 2008. p. 22–4.
[11] Niu G. Data-Driven driven technology for engineering systems health management. 2017. doi:10.1007/978-981-10-2032-2.
[12] Sharp M, Weiss BA. Hierarchical modeling of a manufacturing work cell to promote contextualized phm information across multiple levels. Manuf Lett 2018;15:46–9. https://doi.org/10.1016/j.mfglet.2018.02.003.
[13] Lei Y, Li N, Guo L, Li N, Yan T, Lin J. Machinery health prognostics: a systematic review from data acquisition to RUL prediction. Mech Syst Signal Process 2018;104:799–834. https://doi.org/10.1016/j.ymssp.2017.11.016.
[14] Jennions IK, Niculita O, Esperon-miguez M. Integrating IVHM and asset design. Nternational J Progn Heal Manage 2016;7:1–16.
[15] Niculita O, Nwora O, Skaf Z. Towards design of Prognostics and Health Management solutions for maritime assets. Procedia CIRP 2017;59:122–32. https://doi.org/10.1016/j.procir.2016.10.128.
[16] Aizpurua JI, Catterson V. Towards a methodology for design of prognostic systems. Annu Conf Progn Heal Manag Soc 2015;7069:504–17.
[17] Nordqvist T., Gustafsson M.-L. Systems engineering, architecture frameworks and modelling & simulation n.d.
[18] Vogl GW, Weiss BA, Helu M. A review of diagnostic and prognostic capabilities and best practices for manufacturing. J Intell Manuf 2016:1–17. https://doi.org/10.1007/s10845-016-1228-8.
[19] Elattar HM, Elminir HK, Riad AM. Prognostics: a literature review. Complex Intell Syst 2016;2:125–54. https://doi.org/10.1007/s40747-016-0019-3.
[20] Zhang Z, Si X, Hu C, Lei Y. Degradation data analysis and remaining useful life estimation: a review on wiener-process-based methods. Eur J Oper Res 2018;271:775–96. https://doi.org/10.1016/j.ejor.2018.02.033.
[21] Dumargue T, Pougeon J, Masse J. An approach to designing PHM systems with systems engineering. Eur. Conf. PHM Soc. 2016 Proc. 2016.
[22] Cocheteux P, Voisin A, Levrat E, Iung B. Prognostic design: requirements and tools. 11th Int. Conf. Mod. Inf. Technol. Innov. Process. Ind. Enterp. MITIP; 2009. 2009.
[23] Aizpurua JI, Catterson V. Strathprints institutional repository ADEPS: A methodology for designing prognostic applications. Eur. Conf. PHM Soc. 2016 Proc. 2016:86–100.
[24] Khorasgani H, Biswas G, Sankararaman S. Methodologies for system-level remaining useful life prediction. Reliab Eng Syst Saf 2016;154:8–18. https://doi.org/10.1016/j.ress.2016.05.006.
[25] Liu B, Liang Z, Parlikad AK, Xie M, Kuo W. Condition-based maintenance for systems with aging and cumulative damage based on proportional hazards model. Reliab Eng Syst Saf 2017;168:200–9. https://doi.org/10.1016/j.ress.2017.04.010.
[26] Mao K, Zhu Y, Chen Z, Tao X. A visual model-based evaluation framework of cloud-based prognostics and health management. IEEE Int. Conf. Smart Cloud 2017. p. 33–40. https://doi.org/10.1109/SmartCloud.2017.12.
[27] Yang L, Wang J, Zhang G. Aviation PHM system research framework based on PHM big data center. 2016 IEEE Int. Conf. Progn. Heal. Manag 2016. ICPHM; 2016. p. 1–5. https://doi.org/10.1109/ICPHM.2016.7542824.
[28] Zhang G, Wang J, Lv Z, Yang Y, Su H, Yao Q, et al. A integrated vehicle health management framework for aircraft - a preliminary report. Progn. Heal. Manag. (PHM), 2015 IEEE Conf. 2015. p. 1–8. https://doi.org/10.1109/ICPHM.2015.7245034.
[29] Han S, Yu J, Tang D, Liu H. Implementation and verification of prognostics and health management system using a configurable system of systems architecture. 2016 IEEE Int Conf Progn Heal Manag ICPHM; 2016. https://doi.org/10.1109/ICPHM.2016.7542835. 2016.
[30] Keller K, Peck J, Swearingen K, Gilbertson D. Architectures for affordable health management. AIAA Infotech Aerosp 2010;2010:1–11. https://doi.org/10.2514/6.2010-3435.
[31] Service SC. International Standard 13374-1: Condition monitoring and diagnostics of machines – Data processing, communication, and presentation. Int Organ Stand 2003;2003:1–8.
[32] Swearingen K, Majkowski W, Bruggeman B, Gilbertson D, Dunsdon J, Sykes B. An open system architecture for condition based maintenance overview. 2007 IEEE Aerosp. Conf. 2007. p. 1–8. https://doi.org/10.1109/AERO.2007.352921.
[33] Felke T, Hadden G, Miller D, Mylaraswamy D. Architectures for integrated vehicle health management. AIAA Infotech@aerosp 2010;2010:3433. https://doi.org/10.2514/6.2010-3433.
[34] Committee S., Reliability I. IEEE standard framework for prognostics and health management of electronic systems ieee standard framework for prognostics and health management of electronic systems. 2017. doi:10.1109/IEEESTD.2017.8227036.
[35] International Council on Systems Engineering. Systems engineering handbook - A a guide for system life cycle processes and activities. 2015.
[36] Abramovici M. Smart Product Engineering. In Proceedings of the 23rd CIRP Design Conference. 2013. https://doi.org/10.1007/978-3-642-30817-8. Bochum, Germany.
[37] Thomas E, Ravachol M, Quincy JB, Malmheden M. Collaborative complex system design applied to an aircraft system. Proc. 9th Int. Model. Conf. 2012. p. 855–66. https://doi.org/10.3384/ecp12076855.
[38] BKCASE Editorial Board. Guide to the systems engineering body of knowledge. 2015.
[39] Martin JN. Overview of the EIA 632 standard: processes for engineering a system. Proc. 17th Digit. Avion. Syst. Conf. 1. 1998. https://doi.org/10.1109/DASC.1998.741462. B32-1–9.
[40] Proper E, Op't Land M, Cloo J, Waage M, Steghuis C. Enterprise Architecture - Creating Value by Informed Governance. Springer, Heidelberg. 2009. https://doi.org/10.1007/978-3-540-85232-2.
[41] PLMCC. Catia V6 introduction buggy case study. 2016.
[42] IBM. Systems engineering tutorial for rational rhapsody. 2009.
[43] Levis A, Sage AP, Rouse WB. Handbook of systems engineering and management-system architectures. Handb. Syst. Eng. Manag.; 1999. p. 427–54.
[44] Symposium I, Hutchison D. NASA formal methods. vol. 7871. Minneapolis, MN, USA: Springer International Publishing; 2016. https://doi.org/10.1007/978-3-642-38088-4.
[45] Wang Y, Yu S, Xu T. A user requirement driven framework for collaborative design knowledge management. Adv Eng Informatics 2017;33:16–28. https://doi.org/10.1016/j.aei.2017.04.002.
[46] Wheatcraf L.S. Thinking ahead to verification and validation. 2012.
[47] Voirin JL. Model-based System and Architecture Engineering with the Arcadia Method. Elsevier; 2017 Nov 22. https://doi.org/10.1016/C2016-0-00862-8.
[48] Aguilar RM, Mun V, Noda M, Bruno A, Moreno L. Verification and validation of an

intelligent tutorial system. Expert Syst Appl 2008;35:677–85. https://doi.org/10.1016/j.eswa.2007.07.024.

[49] Maalem S, Zarour N. ScienceDirect challenge of validation in requirements engineering. J Innov Digit Ecosyst 2016;3:15–21. https://doi.org/10.1016/j.jides.2016.05.001.

[50] Poisson P, Chinniah Y, Jocelyn S. Design of a safety control system to improve the verification step in machinery lockout procedures : a case study. Reliab Eng Syst Saf 2016;156:266–76. https://doi.org/10.1016/j.ress.2016.07.016.

[51] Simeu-abazi Z, Di M, Knotek M. Fault diagnosis for discrete event systems : modelling and verification. Reliab Eng Syst Saf 2010;95:369–78. https://doi.org/10.1016/j.ress.2009.11.007.

[52] Bozzano M, Cimatti A, Katoen J, Katsaros P, Mokos K, Yen V, et al. Spacecraft early design validation using formal methods. Reliab Eng Syst Saf 2014;132:20–35. https://doi.org/10.1016/j.ress.2014.07.003.

[53] SAE international Group. APR4754A guideline for development of civil aircraft and systems 4970. 2010.

[54] Standard I. ISO 17359 condition monitoring and diagnostics of machine general guidelines. 2003.

[55] Li R, Verhagen WJC, Curran R. A functional architecture of prognostics and health management using a systems engineering approach. Proc. Eur. Conf. PHM Soc. 2018. p. 1–10.

[56] Chen J, Lyu Z, Liu Y, Huang J, Zhang G, Wang J, et al. A big data analysis and application platform for civil aircraft health management. 2016 IEEE Second Int Conf Multimed Big Data 2016:404–9. https://doi.org/10.1109/BigMM.2016.54.

[57] SKYbrary. Flight data monitoring (FDM) n.d. https://www.skybrary.aero/index.php/Flight_Data_Monitoring_(FDM)#Description.

[58] International Organization for Standardization. ISO 13374-2 condition monitoring and diagnostics of machines-Part part 2: dData processing. vol. 7. 2004.

[59] Jan Dietz, Erik Proper JT. Enterprise architecture at work: modelling, communication and analysis. Dordrecht Heidelberg London New York: Springer; 2000. https://doi.org/10.15713/ins.mmj.3.

[60] Lemazurier L, Chapurlat V, Grossetête A. ScienceDirect an MBSE approach to pass requirements to functional architecture approach to from requirements functional architecture an from requirements to architecture * requirements MBSE approach from to functional functional architecture. IFAC-PapersOnLine 2017;50:7260–5. https://doi.org/10.1016/j.ifacol.2017.08.1376.

[61] Steimer C, Fischer J, Aurich JC. Model-based design process for the early phases of manufacturing system planning using SysML. Procedia CIRP 2017;60:163–8. https://doi.org/10.1016/j.procir.2017.01.036.

[62] Gardan J, Matta N. Enhancing knowledge management into systems engineering through new models in SysML. Procedia CIRP 2017;60:169–74. https://doi.org/10.1016/j.procir.2017.01.052.

[63] Bougain S, Gerhard D. Integrating environmental impacts with SysML in MBSE methods. Procedia CIRP 2017;61:715–20. https://doi.org/10.1016/j.procir.2016.11.196.

[64] Zeigler BP, Nutaro JJ. Towards a framework for more robust validation and verification of simulation models for systems of systems. J Def Model Simul Appl Methodol Technol 2016;13:3–16. https://doi.org/10.1177/1548512914568657.

[65] John Hsu RC. Advances in systems engineering. American Institute of Aeronautics and Astronautics, AIAA; 2016https://doi.org/10.2514/4.104091.