Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Triple V Product Development Framework and Its Interoperability between Product, Model and Data Lifecycles

Qing Li*, Hailong Wei*, Chao Yu*, Shuangshuang Wang*, Linlin Xiao**

**Department of Automation, Tsinghua University, Beijing 100084, China*
(Tel: +86-10-62771152; e-mail: liqing@tsinghua.edu.cn)

***China Industrial Control Systems Cyber Emergency Response Team, Beijing, 100040, PR China*

Abstract: In order to reduce the time and cost of verification, validation and accreditation (VV&A), increase the probability that design will succeed at one time, the traditional product development framework and methodology, which is usually described as the V framework, is extended through the double V framework to the model and data based triple V framework. The new framework provides an opportunity to converge data lifecycle, model lifecycle and product (development) lifecycle together. The paper analyses the interoperability of proposed triple V framework and implements it in a ship damage control system development project.

Keywords: V Framework, System Development, Model, Methodology.

1. INTRODUCTION

Product development is a process that consists of requirement analysis, system design, testing, manufacturing and integration phases and activities. Royce (1970) introduced a linear sequential life cycle process, which was later quoted as "waterfall model" by Bell and Thayer (1976), even though Royce never use this term. There are deep relationships between different phases along the process: design results of upstream activities will be verified and validated in downstream stages. The waterfall process can be redrawn as a V framework in order to embody these relationships.

There is a clear, one-to-one verification, validation and accreditation (VV&A) relationships between the two loops of the V framework, as well as VV&A relationships between two connected phases. That means the design result of the product definition phase cannot get the final verification and confirmation until the product is completed.

In order to avoid defects and flaws of a phase passed to next phase, one primary principle is to complete the detailed review and ensure the correctness before entering the next stage, which increases the time and cost of product and cost. Besides, Because the nature of product development requires constant iteration, it is almost impossible not to pass errors to the next phase.

The paper reviews product development frameworks from V to double V, and presents a model and data driven triple V framework for product development in the big data era. The new framework and its interoperating methods are applied in a ship damage control system design and development project.

2. DOUBLE V AND TRIPLE V FRAMEWORKS

The V framework has undergone continuous improvement since being proposed. Kevin Forsberg and Harold Mooz (1991) implemented system engineering methodology to a traditional V framework and added a new dimension to review the product development process in a multi-dimensional perspective, which can be seen as a rudiment of double V framework. The double V framework emphasizes risk management, concurrent opportunities, VV&A planning, and continuous verification of users.

In order to transfer design results at different phases, digital twins and digital prototypes are established with the help of evolving computer technology and modelling technology. Through modelling and simulation via computer, the design results can be verified before the product's final completion. Model based systems engineering (MBSE) methodology forms the other V of the double V framework presented in this paper, which is shown in Fig. 1. The main features are listed as below.

- MBSE helps change the traditional way of conveying design results from documents and drawings to digital models, and also realize information sharing and reuse along the whole process.
- The traditional tests, experiments, verifications, and validations based on product lifecycle are replaced or improved by digital model-based simulation and digital testing activities in the design specification loop, reducing time and cost of design.
- The digital model can verify and confirm the testing phase by helping find defects and build solutions in the integration loop, which also provides opportunities for the in-depth development of the mechanism research of the product.

Because of the development of information technology and digital model technology, MBSE's double V framework are widely used in engineering practice. The maturity of computer aided technologies and tools simplifies digital model-based design, delivery, validation, implementation and management.

Results of digital analysis get more and more qualitatively and quantitatively consistent with product experimental tests.

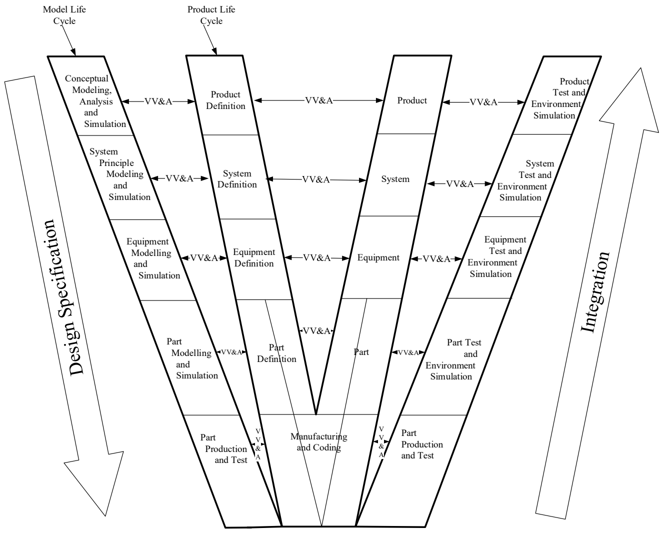


Fig. 1. Model based systems engineering (MBSE) and double V framework

Since MBSE theory is still evolving and under development, the double V framework also faces some challenges. It's hard to establish proper models when comes to a new field lacks of theoretical research. The consistency between abstracted models and the original objects, the simulation and the virtual operation, is not so easy to verify. The emerging big data technology provides a new solution for those problems.

Big data technology is developed based on data science and technology such as data mining, knowledge engineering, etc. , supported by machine learning technology(Amalina et al. 2020). There is a close connection between the product and its model and data. Models express the main features of a product, models and product generate a large amount of data in the whole life cycle, data helps the verification and validation between models and product.

Data-driven machine learning technology, especially deep neural network, has been widely used in the fields difficult to extract features and build models, such as image processing, automatic translation, and speech recognition, which shows the opportunities so solve challenges faced by MBSE double V framework. Volponi (2008) provided a framework of an engine controller with embedded neural network, the mode of unmodeled engine is identified, modelled and compensated through the learning mechanism of neural network.

Combining the advantage of big data technology, a triple V framework of model and data driven product development is presented as shown in Fig. 2. The main principles are listed as below.

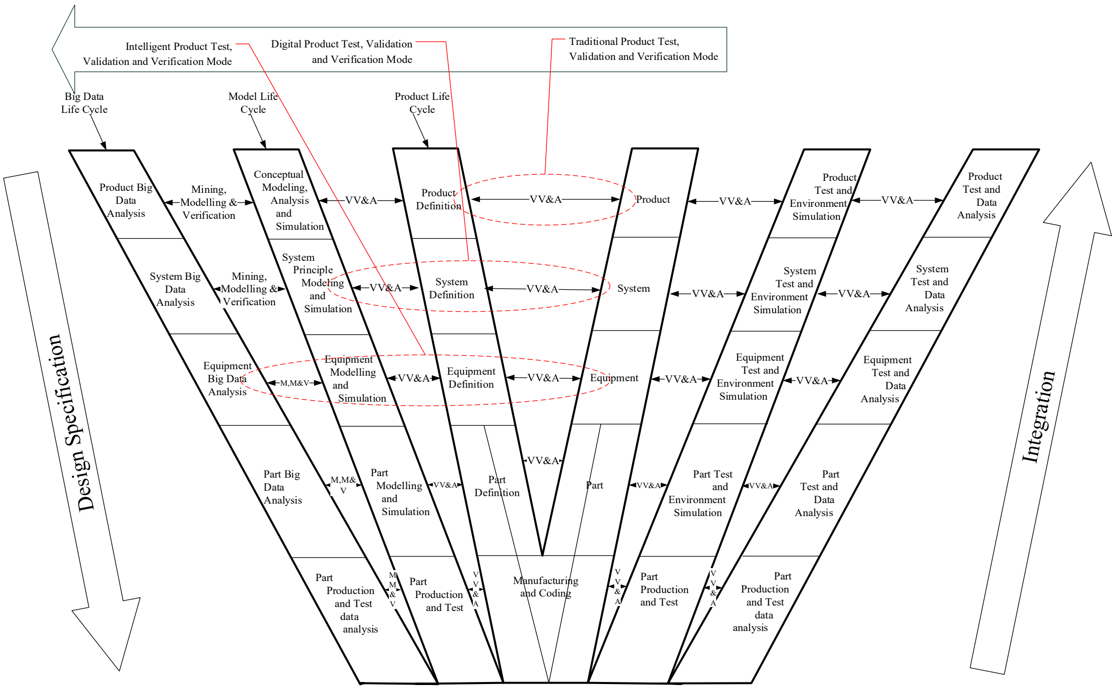


Fig. 2. Model and data driven triple V framework

- In the design specification loop, there is a relationship of mutually support and verification between data, model, and product design. The digital model is built on the basis of understanding product mechanism, and helps verify the simulation and design results. Model-based digital testing and simulation is performed before testing physical product to generate data of the structure, behaviour and performance of the product. In the meantime, analysing relevant data of existing products or models through machine learning technology can support the construction of digital models and the VV&A test.
- In the integration loop, a large number of data is generated by the verification process of part, equipment, systems and products. Comparing the test data of physical product and model simulation result and finding out the reasons for

difference, the quality of the developed product and modelling theory can be improved. Meanwhile, environmental models can provide test environments and arbitrary input for product testing. The black box test can be changed into a white box test on the basis of the understanding of the mechanism of developed product embedded in models, improving analysis and decision-making abilities. Data is always the core in the interaction between product and models.

3. INTEROPERABILITY OF TRIPLE V FRAMEWORK

In order to realize the triple V framework in an engineering environment, multiple dimensions of interoperability shall be considered and realized as shown in Fig. 3.

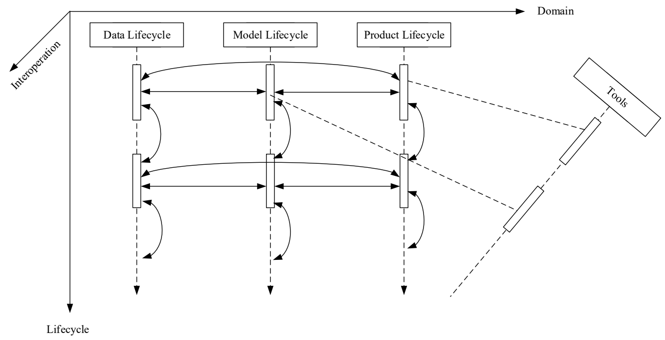


Fig. 3. Interoperability of triple V framework

- Interoperating between different phases of product (development) lifecycle. Based on the traditional systems engineering technology, design results of every phase are transferred to the next phase through paper-based documents. With MBSE, digital model becomes the design tool and design result. In different phases of product development, from the determination of objectives, the definition of requirements, the preliminary design of the system to the design of different subsystems and fields through gradual decomposition, there are huge differences in the modelling methods and tools used due to different design purposes. Because of the relatively mature modelling methods and tools in the detailed design phase, the models in the requirements and system design phase have become the focus of current research. SysML is the main development direction of system modelling language.

- Interoperating between different phases of model lifecycle. With the whole process of product development, different modelling languages are needed in different phases. Zachman, FEAF (Federal Enterprise Architecture), DoDAF (Department of Defense Architecture Framework) and TOGAF (The Open Group Architecture Framework) are involved in a large number of modelling methods along the life cycle dimension. These modelling methods and their mapping and transformation between different phases have always been a research hotspot. In the field of software engineering, model driven architecture (MDA) defines the mapping process from the business model / computation independent model (CIM), platform independent model (PIM), platform specific model (PSM) to code. However, at present, the model mapping methods in various stages are not complete, especially the mapping technology from business model to computing model.

- Interoperating between different stages of data lifecycle. Data has been generated, collected, stored and used along with the whole process of product development cycle. As implied by the EPC (Event Process Chain) modelling method in ARIS (Architecture of Integrated Information System), the relationship between data is mainly formed by the temporal and logic relationship between business activities. Data is the input and output by business activities to support the operation of business activities. With the development of big data and the new generation of artificial intelligence based on data, there is an independent coupling relationship between data in different phases through machine learning and big data processing. The methodology of big data analysis, including CRISP-DM (Cross-industry Standard Process for Data Mining), defines the basic framework of data transformation.

- Interoperating between activities of data lifecycle and model lifecycle: The simulation and operation of the model can generate data, which can support the model construction through pattern recognition and other technologies. With the development of machine learning technology, it is a technical development direction to form a new model through the learning of big data, or to modify the unmodeled mode in the existing model.

- Interoperating between activities of model lifecycle and product (development) lifecycle: Due to the modelling of design results, the integration trend of model life cycle and product development life cycle, and the development of digital twin technology, a solution of deep integration of model and physical product design is being built.

- Interoperating between activities of data lifecycle and product (development) lifecycle: A lot of data will be generated in the process of product experiment, test and use, and the existing data can play a role of benchmark and reference for product test and tracking. Internet of things technology and health management technology based on Internet of things make data collection and processing an integral part of intelligent products.

- Interoperation between different tools: In order to support the engineering practice of MBSE, a large number of modelling techniques and tools have been developed, including SysML, Modelica in modelling technology, Harmony, SYSMOD, FAS, OOSEM in MBSE method, DoDAF and UAF in architecture. To support the application of SysML in product design, MagicGrid provides a methodology for model transformation.

In general, due to the immature transformation solution of product development process supported by MBSE, and the immature integration solution of big data technology and product development process, the interoperability of various dimensions of triple V framework and related tool solutions are still in the process of development.

4. ENGINEERING IMPLEMENTATION IN A SHIP DAMAGE CONTROL SYSTEM DESIGN

4.1 Case Introduction

The damage control system (DCS) improves the damage control capability of ships. DCS is a system for information retrieval and equipment monitoring, which supports crew to detect, analyse and deal with various harmful conditions to ships (Calabrese et. al. 2012). Typical situations include firefighting, anti-sinking and leaking. Based on the triple V framework, this section discusses the design workflow and interoperating methods of a DCS that meets fire suppression requirements.

4.2 Structural Approach of Design Specification

The main body of the triple V framework is product (development) lifecycle, model lifecycle and data lifecycle, aiming to improve the design and verification speed while reducing R&D time and costs. For ease of analysis, this section only describes how to better complete the product definition phase with models and data, and uses Rational Rhapsody for modelling analysis.

Step 1. Capture product requirements.

The requirement diagram of SysML is used to build the DCS requirement model and map the requirement to the use cases via use case diagram of SysML. For ease of analysis, this section only discusses the firefighting use case for DCS, with the captain, actuators and sensors as actors.

Step 2. Verification of product definition phase

Through literature research and interviews with ship crew, the simplified ship firefighting process with DCS is shown in Appendix A. (JT 558-2004). In order to define the DCS more accurately, it is necessary to describe the use case activity diagram (black box) and the use case scenario based on the block definition diagram and the sequence diagram of SysML from the firefighting use case, and generate an executable ship state machine diagram by referring to the actual operation of the DCS in the ship. The state machine diagram verifies the feasibility of the conceptual model through simulation. The state machine diagram of the ship with DCS is shown in Fig. 4. The state jump is triggered by specific events or messages from actors.

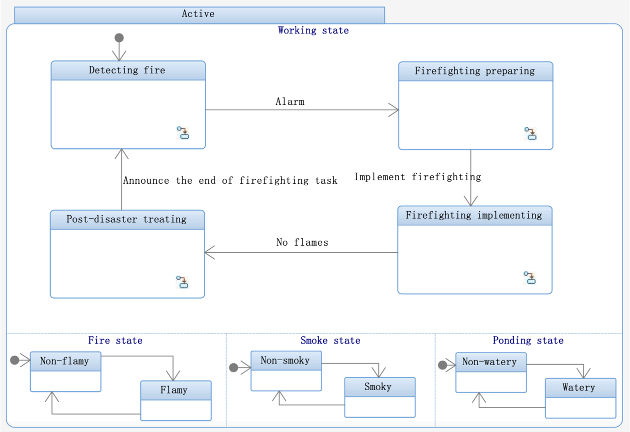


Fig. 4. State machine diagram of the ship with DCS

Starting the state simulation in Rhapsody, the state jumps within the four parallel states of "Working state", "Fire state", "Smoke state", "Ponding state" triggered by different events and messages. Therefore, it can be verified that the DCS equipped on the ship meets the design expectations, and theuser requirements can be satisfied to complete the firefighting task.

Step 3. Validation of product definition phase

Usually, fires occur quickly and would cause large losses. Therefore, the time it takes to complete a firefighting mission determines the effectiveness of the DCS function. The effectiveness can be measured by comparing the probability of successful fire suppression and the average length of time cost with DCS and without DCS. In order to obtain an accountable result at the beginning of the design, the models built above need more refined simulation driven by data.

1) Fire feature simulation based on PyroSim

PyroSim is a graphical software for firefighting dynamic simulation developed on the basis of firefighting simulation software FDS (Long et. al. 2017). The internal structure of the ship is complex and the combustibles are diverse. To simplify the problem, this study constructed a 3-storey, 24-room structure of the ship's main deck cabin.

According to an autologous experiment conducted by the British experimental physiologist Blagdon in 1775, even 48~50 C° will quickly cause harm to the human body in a humid environment (Wang 2007). Since the ship cabin is in a humid environment, 50 °C is selected as the temperature to damage the human body in the simulation. When the temperature reaches 50 °C, the area is regarded been damaged by the fire.

Setting the temperature having reached 50 °C as the benchmark, the percentage of the damaged area of the first floor after somewhere caught fire is calculated over time. Starting the simulation in PyroSim, in the absence of firefighting operations, the percentage of the damaged area of the first floor reached 100% at approximately 120 s, and the firefighting mission failed.

According to Fan(1995), the heat release rate of the fire source during the fire process is proportional to the square of the time. Assume that the percentage of the area damaged by the disaster is also approximately proportional to the square of time, as the function below:

$$H(t) = \begin{cases} 6.944 \times 10^{-3} \times t^2, t < 120; \\ 100, t \geq 120. \end{cases} \tag{1}$$

In which, H represents the value of the percentage of the damaged area, and its value range is [0, 100]. When H=100, the firefighting task fails; *t* represents the burning time, and the unit is *s*.

2) Compare the ship firefighting effectiveness with DCS and without DCS

The ship firefighting process without DCS is shown in Appendix B. Comparing Appendix A. and Appendix B., when the ship is equipped with DCS, since the DCS can display the firefighting status in real time, the process of repeatedly reporting the information is reduced, and subtasks are allowed to be executed concurrently, thereby making interoperations between different departments fast and close.

The executable state machine diagrams of the ships with or without DCS were constructed to study the influence of DCS on the firefighting task. There are other subsystems or officers that are required to perform the firefighting capability when the DCS is not assembled. That is, with or without DCS, the ship firefighting process can be represented by the jump of the same four sub-states of the "Working state" of the ship. The specific operations of a state are defined internally, as shown in Fig. 5.

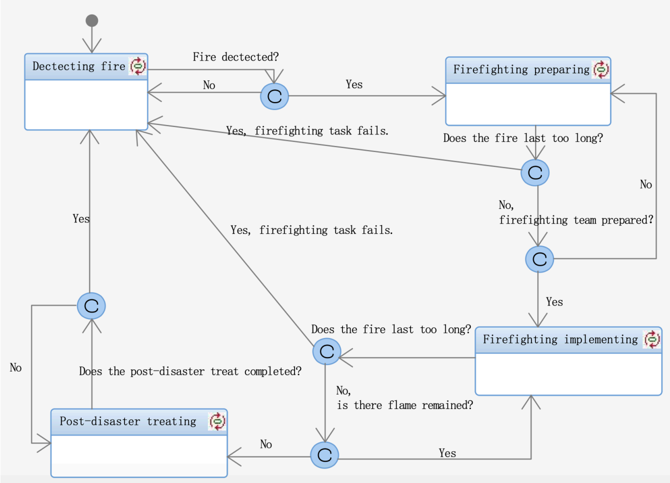


Fig. 5 Data-driven state machine simulation model

The "detecting of fire" mainly depends on the hardware equipment such as the fire alarm controller installed on the ship. Therefore, the time to detect fire and the probability of successful detecting are the same regardless of whether or not the DCS is installed. The "firefighting preparing" process and the "post-disaster treating" process require information exchange between different regions and different individuals. The information integration capability of DCS can effectively reduce the interaction time and increase the probability of success. The probability of successful firefighting needs to consider the time the fire has lasted for. According to the PyroSim fire feature simulation, the longer the burning lasts, the larger the area of the damaged area would be, and the smaller the probability of successful firefighting would be. Therefore, the following assumption is made:

$$p_{\text{implement}} = 1 - H/100. \tag{2}$$

Combining equations (1) and (2), there are:

$$p(t_{\text{total}}) = \begin{cases} 1 - 6.944 \times 10^{-5} \times {t_{\text{total}}}^2, t_{\text{total}} < 120; \\ 0, t_{\text{total}} \geq 120. \end{cases} \tag{3}$$

$$t_{\text{total}} = l \times t_{\text{detect}} + m \times t_{\text{prepare}} + n \times t_{\text{implement}} \tag{4}$$

In which, l, m and n respectively indicate the number of times staying in "Detecting fire", "Firefighting preparing" and "Firefighting implementing" states before the successful state jumps.

The state machine diagrams of the ships with DCS and without DCS are simulated 1 000 times respectively. The number of successful firefighting is 940 times when the DCS is assembled, so the firefighting success rate is 94%. While the DCS is not assembled, the number of successful firefighting is 806 times, so the firefighting success rate is 80.6%. The comparison of the histograms of the total firefighting time of ships with and without DCS is shown in Fig. 6. The total firefighting time of ship with DCS is shorter than the total firefighting time of ship without DCS, and is more concentrated in distribution, which validates the effectiveness of DCS.

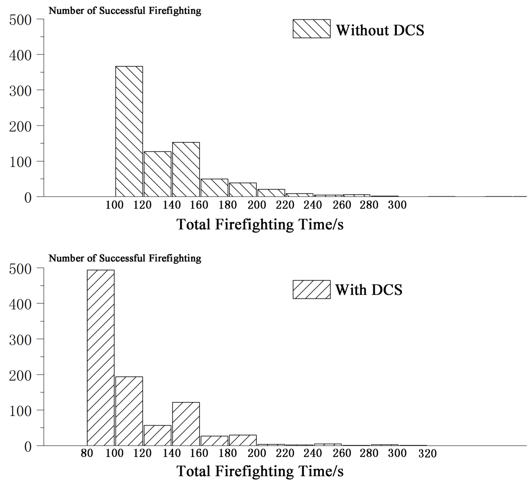


Fig. 6. Histograms of the total firefighting time

Step 4. Accreditation and extension of product definition phase

The accuracy of the product definition is verified based on the conceptual modelling and simulation. The data-driven refined simulation can validate the effectiveness of the defined product to meet the user's requirements. With the help of model and data, the product definition phase is accredited and the product design and development process turns to the next phase.

The initial requirement for DCS is to put out fire as soon as possible, but during the analysis it was thought that reducing the casualty rate is the more important goal, that is, the key is to ensure that people are far away from the fire. The scene of the fire is often accompanied by heavy smoke, so the visibility is very low. If DCS could provide escape route planning function, it should be possible to avoid personnel blockage and maximize the use of the escape route. Based on the cabin model built by PyroSim in the step 3, the fire escape model is defined with PathFinder software: The target of escape is to enter the upper deck. There are 24 people on the first floor, 40 people on the second floor, and 20 people on the third floor. The people on each floor are randomly distributed. When a fire occurs, people generally move quickly. When moving, the moving speed of the person walking is 1.8m/s.

Escape route planning was carried out to effectively balance the number of people escaped from both sides of the stairs, thus making full use of the escape routes on both sides. The simulation result shows that Route planning can save 8.7% of escape time. Therefore, having the route planning function should be added to the product requirement of DCS, which shows the iteration of product design with the help of model and data.

The interoperation between product, model, and data is the key to implement the proposed triple V framework. Through model life cycle and big data life cycle, it can verify the accuracy of product design and ensure effectiveness of satisfying users' requirements in the early stage of product design and development process. In addition, data-driven model simulation can in turn help designers improve product design, complement possible missing requirements, and make better decisions.

5. SUMMARY AND CONCLUSIONS

This paper proposed a model and data driven triple V framework for complex product development, combining the advantages of MBSE and data mining and analysis technology, integrating product (development) life cycle, model life cycle and big data life cycle. The paper verified the proposed framework by implementing it to the design of the firefighting subsystem of a ship damage control system. Through interoperating processes of data, model and product design, new data, new model and new design results are continuously produced and mutually verified.

The triple V framework includes multiple dimensions of interoperability. Modelling methods, model mapping methods and tools are needed to support the realizing and implementation of the new framework, which are still researched and developed.

ACKNOWLEDGEMENTS

This work is sponsored by the National Natural Science Foundation of China, No. 61771281, the 2018 Industrial Internet innovation and development project.

REFERENCES

Amalina, F., I. A. Targio Hashem, Z. H. Azizul,et al. (2020). Blending Big Data Analytics: Review on Challenges and a Recent Study." IEEE Access 8:3629-3645.

Bell, Thomas, and T. Thayer. (1976). Software Requirements: Are They Really a Problem?. Proceedings of the 2nd international conference on Software engineering.

Calabrese F, Cataldo M, Corallo A, et al. (2012). Damage control system: an application for ship safety and security. Proceedings of the 9th IFAC Conference on Manoeuvring and Control of Marine Craft. Arenzano, Italy: IFAC, 2012: 103-108.

Fan W C. (1995). Concise tutorial on fire science. Anhui: University of Science and Technology China Press, 1995. (in Chinese)

Forsberg K., Mooz H. (1991). The relationship of system engineering to the project cycle. Proceedings of the First Annual Symposium of National Council on System Engineering. October 1991: 57–65.

Long X F, Zhang X Q, Lou B. (2017). Numerical simulation of dormitory building fire and personnel escape based on Pyrosim and Pathfinder. Journal of the Chinese Institute of Engineers, 2017, 40(3): 257-266.

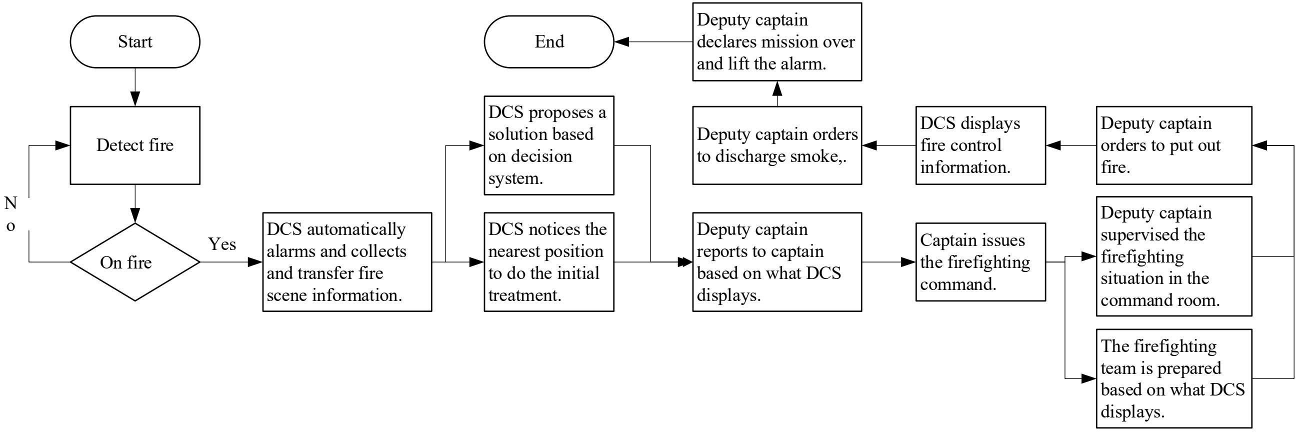
Ministry of transport of the people's republic of china. (2004). JT 558-2004: Due process of on-board fire drills. (in Chinese)

Royce, W. (1970). Managing the development of large software systems. IEEE Wescon, August 1970.

Volponi A. (2008). Enhanced self-tuning on-board real-time model (eSTORM) for aircraft engine performance health tracking. NASA/CR—2008-215272, July 2008

Wang Z J. (2007). The concept of constant internal environment - the great contribution of Berna. Bulletin of Biology, 2007, 42(9):56-57. (in Chinese)

Appendix A. Simplified ship firefighting process with DCS



Appendix B. Simplified ship firefighting process without DCS