



# Study on Top Architecture Design of Nuclear Power System Based on an Incremental Iteration Generation Method

Pan Xinxin<sup>(✉)</sup>, Song Chunjing, and Ming Yao

Shanghai Nuclear Engineering Research and Design Institute Co. Ltd, Shanghai, China  
{panxinxin, songcj, mingyao}@snerdi.com.cn

**Abstract.** Nuclear power system is a complex system with strict requirements, various functions and huge structure. Its multi-disciplinary complexity and systematic correlation make it necessary to implement the top-down forward design concept in the nuclear field. It is of great significance to establish a set of top-down design flow with the model based on requirements to guide the research and design activities of new reactors.

Systems engineering involves two aspects: technology and management. From the technical point of view, it is a demand-driven process that successively carries out functional and logical analysis, and finally develops products that meet user needs through repeated integration and verification. From the management perspective, it mainly includes many management topics such as plan management, requirement management, risk management and status management for the whole process of technology. Systems engineering is a universal scientific method for organizing and managing systems.

In order to support the top-down forward design of the top architecture of nuclear power system, an incremental iterative generation method of functional architecture and logical architecture driven by models with the MBSE (model based system engineering) method was proposed. Firstly, based on the existing knowledge of normal conditions, hierarchical functional analysis was carried out to establish the functional structure of nuclear power system. Secondly, based on the definition of system function and logical structure, the system logical component set was obtained through functional-logic mapping to form the definition of nuclear power system structure module, and the internal association of the system was constructed by incremental iteration with the top condition as the traction, and the system architecture scheme was obtained. Finally, the new design requirements were captured by the analysis of the system structure fault mode, which lead to the definition of abnormal conditions and entered the next functional iteration. Until the requirements converged, a complete nuclear power system architecture covering normal and other conditions was formed. The proposed method was verified by the application of a certain reactor design.

**Keywords:** Nuclear power system · Architecture generation · Abnormal condition · Incremental iteration · Functional-logic mapping · MBSE

## 1 Introduction

In the preliminary design of nuclear power system, key parameters must be demonstrated in accordance with the system logic to obtain the optimal solution [1]. Chen et al. [2] designed the Hualong No.1 emergency core cooling system based on the system engineering method, established the system performance evaluation method with the goal of optimizing the overall performance parameters of nuclear power plant, and obtained the optimal design scheme of emergency core cooling system by using the analytic hierarchy method.

Guided by the system engineering method, SNERDI divided the design process into demand analysis, requirement analysis, system design and design verification, and carried out the design of seismic automatic shutdown system in an orderly manner, which ensured the high reliability of the system and verified the application prospect of the method [3].

In China, the design and management of the nuclear power projects roughly reflected certain system engineering idea, but in the process of the existing research and design of new reactors, the source and selection of the key technologies are often lack of top-level thinking, which will easily lead to system interactive contradiction and system design redundancy and etc.[4]. Systems engineering has the application conditions in the field of nuclear engineering, but the general engineering design is still relatively lacking [5].

With the rise of model-based system engineering (MBSE), guided by MBSE modeling methodology represented by IBM Rational Harmony SE and Dassault MagicGrid, MBSE is increasingly being used in aerospace, automotive and marine applications. Many successful applications show that this method can effectively deal with the challenges facing the top-level design of the complex systems and is helpful for improving the nuclear power system design quality and efficiency. Using MBSE to guide the top-level design in nuclear engineering field has become the inevitable trend in the propulsion system engineering, therefore, developing MBSE is very necessary in the nuclear field [6].

However, it must be pointed out that MBSE is still in the preliminary exploratory stage for the general design application in nuclear power systems. The application of MBSE still needs to solve the problem of integration with specific fields. The MBSE engineering practices in aerospace related fields show that the current difficulty in MBSE application lies in the system modeling method and specific landing scheme combined with the development task [7]. Modeling customization for the industry, especially the realization of system engineering data integration and toolsets, is the key to the successful implementation of MBSE.

To build the top architecture design of the nuclear power system, starting from the known knowledge of operating conditions, the data structure of the function and logical architecture is proposed, and the constraint rules of these structures are defined. Based on the function analysis and logic design iteration, the functional requirements are transferred to the system composition and internal structure, and then the nuclear power system functional architecture and logical architecture are obtained. The study provides a foundation for the application of systems engineering in nuclear power system design.

## 2 Method Introduction

The technical route of this method is shown in Fig. 1. Taking the normal operating condition of nuclear power system as the reference input, through the process conditions  $\rightarrow$  function  $\rightarrow$  logic  $\rightarrow$  fault  $\rightarrow$  conditions, the closed-loop iteration of nuclear power system design is realized, and the system functional architecture and logic architecture are incrementally expanded and improved in a spiraling manner driven by operating conditions, and the completeness of the system design architecture is continuously improved after n-round design iteration.

In each design iteration (from 0 to n), the generation of nuclear power system architecture can be divided into two phases: functional architecture design and logical architecture design. In the initial functional architecture design stage, the hierarchical functional decomposition is firstly carried out for the normal operating conditions, which are obtained based on demand analysis or the existing experience. After functional decomposition, the system function set meeting the requirements of normal operating conditions is obtained, and the hierarchical tree functional architecture of the nuclear power system is formed.

In the initial logical architecture stage, starting from the underlying functions of the system functional architecture, the initial logical component set of nuclear power system is generated by functional-logical component mapping. After the modular combination of logic components set, the initial logic composition of nuclear power system is obtained. On the basis of the initial structure, by defining the failure modes of the related structures, the top-level functional requirements for handling the related faults are derived, which describe the abnormal operating conditions of the nuclear power system and trigger the next iterative update of the functional and logical architecture until the functional requirements converge, at which time the logical composition of the nuclear power system is stable.

When defining the logical structure association, the internal association of the logical module structure under the operating condition is designed incrementally with the operating condition as the traction again, and the complete logical architecture of the nuclear power system covering normal and abnormal operating conditions is generated.

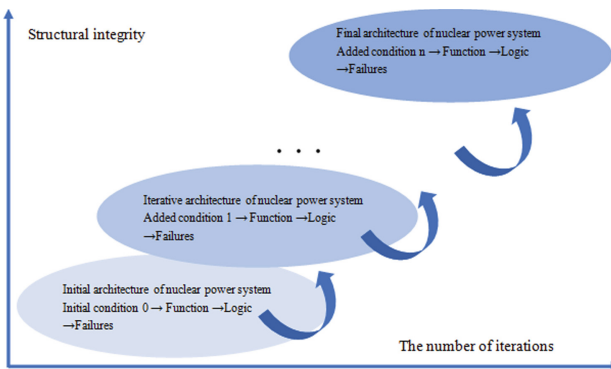


Fig. 1. Iterative generation method for nuclear power system architecture

### 3 Functional Architecture Generation for Nuclear Power System

A functional architecture is a set of functions and their sub-functions that define the transformation of input flows to output flows that the system performs to fulfill its tasks [8] and describes how functions work together to perform system tasks. Complex function can be decomposed into several sub function sequence [9], and continue to decompose the sub-functions until they are no longer separated, forming a hierarchical functional architecture as shown in Fig. 2 [10], which can be described by a tree structure, the root node of the tree is the total function at the top, if the total function has been decomposed, the function at the leaf node position of the function tree is non-indivisible function. The goal of functional analysis of nuclear power system is to obtain the functional architecture for the top-level design of nuclear power system through functional decomposition.

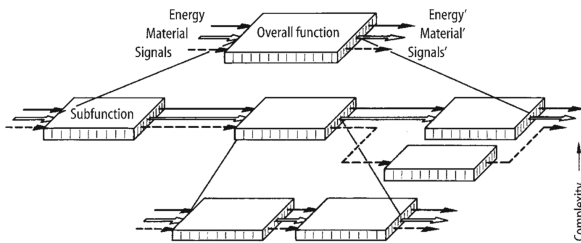


Fig. 2. An example of a hierarchical functional architecture

According to requirement analysis or experience knowledge of nuclear power system, the normal operating conditions of nuclear power system are predefined as reactor start-up, power operation, reactor shutdown and refueling. According to the function definition, a function is considered to be used for energy flow, material flow and signal flow transformation, and a transformation from a particular input flow to an output flow is corresponded to operating conditions. Based on the initialization of the operating conditions of nuclear power system, the top-level functions are shown in Formula (1) and Table 1, wherein  $f_1, f_2, f_3, f_4$  correspond to the top level functions of reactor start-up, power operation, shutdown and refueling, respectively. The functional specifications are listed in Table 1.

$$F = \{f_1, f_2, f_3, f_4\} = \{f_{startup}, f_{powerOpr}, f_{shutdown}, f_{refuel}\} \tag{1}$$

After the above functions are decomposed hierarchically by system designers at the reactor layer, the results are as follows:

$$Arch_F = \{T_{f1}, T_{f2}, T_{f3}, T_{f4}\} = \{T_{startup}, T_{powerOpr}, T_{shutdown}, T_{refuel}\} \tag{2}$$

Among them,  $T_{f1}, T_{f2}, T_{f3}, T_{f4}$  correspond to the tree function structure formed after decomposing functions of reactor start-up, power operation, shutdown and refueling respectively. Each node in  $T_{fi} (i = 1, 2, 3, 4)$  is a system function, and the root node of  $T_{fi}$  is  $fi, T_{fi}.Root = fi$ .

**Table 1.** Top-level function definition based on operating conditions

Function	Name(gerund/phrasal verb)	Functional specifications
$f_{startup}$	Reactor start-up	Start up a nuclear reactor
$f_{powerOpr}$	Power operation	Operating at a specific power, stable external energy supply
$f_{shutdown}$	Shut down	Shut down a nuclear reactor
$f_{refuel}$	Refueling	Refuel

In order to unify the four tree functional structures  $T_{f1}, T_{f2}, T_{f3}, T_{f4}$  into a single functional architecture, on the basis of the root node  $f_i$  of  $T_{fi}$ , the general function  $F$  is defined according to the Formula (1), and the sub-functions of  $F$  are  $f_1, f_2, f_3, f_4$ , and the function tree  $T_F$  is constructed with  $F$  as the root node.  $T_{f1}, T_{f2}, T_{f3}, T_{f4}$  are the four subtrees of  $T$ . Therefore, the functional architecture  $Arch_F$  in Formula (2) can be expressed as:

$$Arch_F = T_F = \{T_{f1}, T_{f2}, T_{f3}, T_{f4}\} = \{T_{startup}, T_{powerOpr}, T_{shutdown}, T_{refuel}\} \quad (3)$$

## 4 Logical Architecture Generation for a Nuclear Power System

### 4.1 Functional and Logical Component Definitions

Assume the function set  $F_{leaf} = \{f_{leaf1}, f_{leaf2}, \dots, f_{leafn}\}$  of the leaf nodes in the function tree  $T_F$ , the function  $f_{leafi} \in F_{leaf}$ . Theoretically, the function can correspond to multiple logical structures which can realize the function. Similarly, each logical structure corresponds to multiple system functions. Therefore, the function of nuclear power system can be defined as below:

$$\begin{aligned}
 &function = (In, Out, Sub, LS, R) \\
 &In = \{in_1, in_2, \dots, in_m\}, Out = \{out_1, out_2, \dots, out_n\}, \\
 &Sub = \{sub_1, sub_2, \dots, sub_p\}, LS = \{ls_1, ls_2, \dots, ls_q\}, \\
 &R = \{r_1, r_2, \dots, r_s\} \\
 &m, n \geq 1, p, q, s \geq 0
 \end{aligned} \quad (4)$$

wherein, function is expressed as five-tuple structure, including the input flow sets  $In$ , the output flow sets  $Out$ , the corresponding subfunction set  $Sub$ , the logical component set  $LS$  realizing the function, and the set  $R$  describing the association between subfunctions. function should have at least one input flow and one output flow, when the component realizing the function is unknown, the logical component sets in Formula (4) is null. However, for the function set of leaf nodes in the above functional architecture, the logical component set  $LS$  of any function is non-null. Therefore, the constraint rules for functional architecture  $Arch_F$  are as follows:

Rule 1: The criterion for the end of function decomposition is that the logical component set  $LS$  for all leaf node functions in the function tree  $T_F$  of the architecture  $Arch_F$  is not null;

Rule 2: For non-leaf node functions in the function tree  $T_F$  of the architecture  $Arch_F$ , its sub-function attribute should not be null,  $f.Sub \neq \emptyset$ .

In addition, according to the definition of the tree, each non-root node in the tree should only have one parent node, so the rules for the functional architecture are as follows:

Rule 3: No two functions of a functional architecture  $Arch_F$  can contain the same subfunctions, that is, for any two functions  $f_i, f_j, f_i.Sub \cap f_j.Sub = \emptyset$ .

Based on the functional architecture definition, to achieve the convergence of functional decomposition, the system must define underlying functions corresponding logical components, and the leaf node functions in the structure tree are stop points, otherwise, it will be unable to find a feasible solution, expressed as:

Rule 4:  $\forall f_i \in F, f_i.LS \neq \emptyset$ , for  $fun \notin F$  and  $fun.LS = \emptyset$ . By function decomposition to obtain functional architecture  $Arch_{fun}$ , the root node of  $Arch_{fun}$  is  $fun$ , and the leaf node function  $Fun_{leaf} = \{fun_{leaf1}, fun_{leaf2}, \dots, fun_{leafn}\}$ . If  $\forall fun_{leafi} \in Fun_{leaf}, fun_{leafi}.LS \neq \emptyset$  the function  $fun$  has feasible functional solution;

Similar to the function, the logical structure also presents a hierarchical tree structure. The logical structure  $L$  of the nuclear power system is defined as the following five-tuple:

$$\begin{aligned}
 L &= (In, Out, Sub, Fun, R) \\
 In &= \{in_1, in_2, \dots, in_m\}, Out = \{out_1, out_2, \dots, out_n\}, \\
 Sub &= \{sub_1, sub_2, \dots, sub_p\}, \\
 Fun &= \{fun_1, fun_2, \dots, fun_q\}, \\
 R &= \{r_1, r_2, \dots, r_s\} \\
 m, n &\geq 1, p, q, s \geq 0
 \end{aligned}
 \tag{5}$$

The logical structure  $L$  is represented as the input flow set  $In$  and output flow set  $Out$  of the structure, and the corresponding internal logical structure constitutes  $Sub$ . The realized function set  $Fun$  and the structure within the  $Sub$  set are associated with the set  $R$ , and there is at least one input flow and one output flow. If  $L$  does not include other structures,  $Sub$  is null. If  $L$  is not associated with any function, the function set  $Fun$  in Formula (5) is null. Obviously, logical structure  $L$  can also use tree to represent its overall architecture  $Arch_L$ , and the constraint rules for logical structure are as follows:

Rule 5: The input and output flows of logical structure  $L$  shall cover all the input and output flows of the function it implements, i.e.:

$$\begin{aligned}
 &\text{If } L.Fun \neq \emptyset, \forall fun \in L.Fun, \\
 &fun.In \subseteq L.In, fun.Out \subseteq L.Out
 \end{aligned}$$

Obviously, if  $L$  implements the function  $fun$ , then  $fun$ 's input flow set and output flow set should be subsets of  $L$ 's input flow set and output flow set, respectively.

Architecture describes the constituent elements and their relationships [11], which are included in both functional architecture and logical architecture definitions. For a functional architecture that has been constructed and has a stable structure, there must

be associations between its sub-functions; otherwise, the architecture is unstable. The logical structure is the same, so the constraint rules are as follows:

Rule 6: If the functional structure  $F$  is stable, there must be associations between its sub-functions, that is, if  $Card(F.Sub) > 1$ , then  $Card(F.R) > 0$  and  $Card(F.R) \geq Card(F.Sub) - 1$ .

Rule 7: If the logical structure  $L$  is stable, there must be associations between its sub-structures, that is, if  $Card(L.Sub) > 1$ , then  $Card(L.R) > 0$  and  $Card(L.R) \geq Card(L.Sub) - 1$ .

In summary, the functional definition contains the corresponding implementation structure, and the logical structure definition contains the corresponding function, so as to maintain the implementation association between the function and the logical structure. Functions can be realized by multiple logical structures, and the logical structures can also realize multiple functions. The relationship between functions and logical structures is many-to-many mapping.

## 4.2 Logical Structure Inference of Nuclear Power System Based on Functional-Logical Mapping

According to the above definition, based on the nuclear power system requirements analysis or known operating conditions after functional decomposition to obtain the functional architecture  $Arch_F$ , the initial logical structure of the nuclear power system  $L_{init}$  can be inferred to calculate as follows:

$$\begin{aligned} L_{init} &= (In, Out, Sub, Fun, R) \\ Arch_F.F_{leaf} &= \{fun_{leaf1}, fun_{leaf2}, \dots, fun_{leafn}\} \end{aligned} \quad (6)$$

Then:

$$\begin{aligned} L_{init}.In &= fun_{leaf1}.In \cup fun_{leaf2}.In \cup \dots \cup fun_{leafn}.In \\ L_{init}.Sub &= fun_{leaf1}.LS \cup fun_{leaf2}.LS \cup \dots \cup fun_{leafn}.LS \\ L_{init}.Fun &= Arch_F.Root \\ R &= \emptyset \end{aligned}$$

On the basis of the initial logical structure  $L_{init}$ , the substructure to which  $L_{init}$  belongs can be divided into modules and the logical structure can be reconstructed, but the logical structure reconstruction task needs to be manually completed. The association set  $R$  between the structures is empty at this time and needs to be defined in a subsequent design session.

## 4.3 Definition of Faults in the Logical Structure of Nuclear Power Systems

The initial logical structure of the nuclear power system is obtained from the analysis of normal operating conditions, which means that the structure can only meet the functional needs of the system under normal operating conditions. In order to deduce abnormal operating conditions of nuclear power system, fault mode information can be introduced for the logical structure, and the new functional requirements for handling logical structure faults can be iterated out, which are called feedback requirements

or feedback operating conditions. Therefore, the definition of logical structure will be upgraded from Formula (5) to:

$$L = (In, Out, Sub, Fun, R, Fau) \tag{7}$$

wherein,  $Fau = \{fau_1, fau_2, \dots, fau_t\} (t \geq 0)$  is the fault mode set of logical structure. According to each fault mode, a specific operating condition can be derived. For example, a pipeline leakage fault may occur, and the system will generate a pipeline leakage operating condition accordingly. Thus, new functional decomposition is started, and the next design iteration is entered until no new functionality is generated and the corresponding logical structure is obtained. The related conceptual entities and their relationships are shown in Fig. 3.

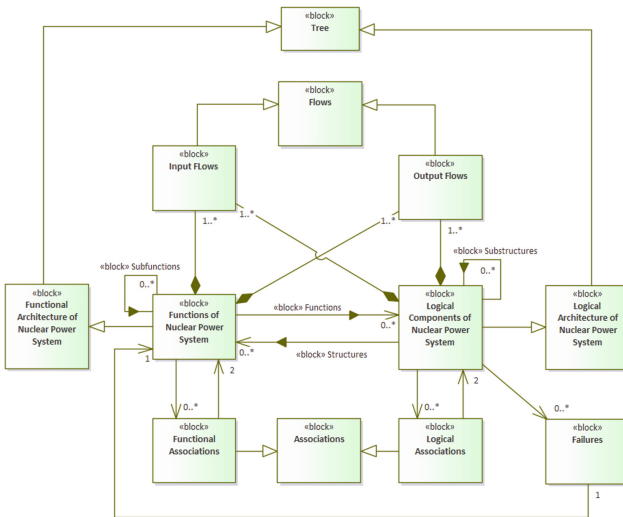


Fig. 3. Conceptual model of function and logical structure of nuclear power system

#### 4.4 Internal Association of Logic Structure of Nuclear Power System

The logical structure derived from the functional architecture only contains internal components and the internal components are not clearly defined, and the internal associations of the logical architecture need to be further defined. In order to reduce the complexity of design, the internal association of logical structure should be completed step by step.

From the definition of function and logical structure, it can be seen that for the operating conditions, functions and logical structures, the associative index path is “top-level functions → operating conditions → ... Non-leaf-node intermediate function... → Leaf-node function set → logical structure set” and the steps to define the internal association of the logical structure based on the operating conditions are as follows:



- 1) Based on logical architecture  $L$ , the logical structure set of leaf nodes is  $L.Sub$ . The function set  $F$  of the operating conditions in which  $L.Sub$  participates is obtained by using the function attributes in the structure definition.
- 2) The function set  $F$  of operating conditions is traversed. For the function item  $f$ , the tree traversal algorithm is used to find out the leaf node function set under  $f$  in the functional architecture, and the corresponding logical structure set  $L_f$  is obtained based on the function-logical structure mapping,  $L_f$  is a subset of  $L.Sub$ .
- 3) For the set  $L_f$ , manually define the composition relationship of the logical structure in the set.

When all operating conditions and functions are traversed, the internal association of the logic structure of the nuclear power system is also defined, thus forming a complete logic structure of the nuclear power system.

## 5 Case Study: Normal Residual Heat Removal System of Nuclear Heating Reactor

The normal residual heat removal system(RNS) of a nuclear heating reactor is taken as the generating object. Starting from the normal operating conditions of heating reactor (start-up, power operation, shutdown and refueling), after the functional architecture of heating reactor is obtained, the logical architecture of normal residual heat removal system of the heating reactor is generated, and the architecture iteration of normal residual heat removal system of the heating reactor based on structural fault feedback is demonstrated.

### 5.1 Functional Decomposition of Heating Reactor Under Normal Operating Conditions

Based on the normal operating conditions of the heating reactor, the corresponding functions are initialized, and the four functions of start-up, power operation, shutdown and refueling are obtained here, and an additional function top-level function – heat supply is created, with heat supply as the root node, the rest of the functions are child-nodes, and the designers perform functional decomposition for 4 child-nodes, and obtain a complete functional architecture tree after hierarchical decomposition of all functions. The initial top-level functional architecture under normal operating conditions is shown in Fig. 4. Figure 5 shows the functional decomposition diagram of the function of “controlling the water quality of primary loop” under startup conditions, and other functions are decomposed in the similar ways.

On the basis of the functional architecture tree, the leaf node function of the tree is assigned to the logical structure, and after the logical structure allocation mapping of the leaf node function of the functional architecture, a logical structure set of the heat supply function of the nuclear heating reactor will be obtained, and these logical structures will be distributed according to different systems, and the initial logical structure of the normal residual heat removal system will be obtained in Fig. 6. On the basis of the initial logical structure, define the logical module of the normal residual heat removal system, and create the composition association between the logic module and the logical

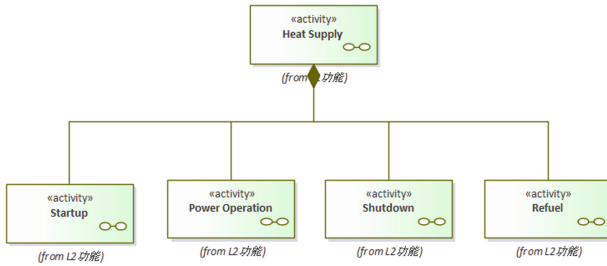


Fig. 4. Initial functional architecture of the heating reactor

structure of each initial allocation, describe which logical components are composed of the logical components of the normal residual heat removal system, and complete the logical structure definition of the normal residual heat removal system, as shown in Fig. 7.

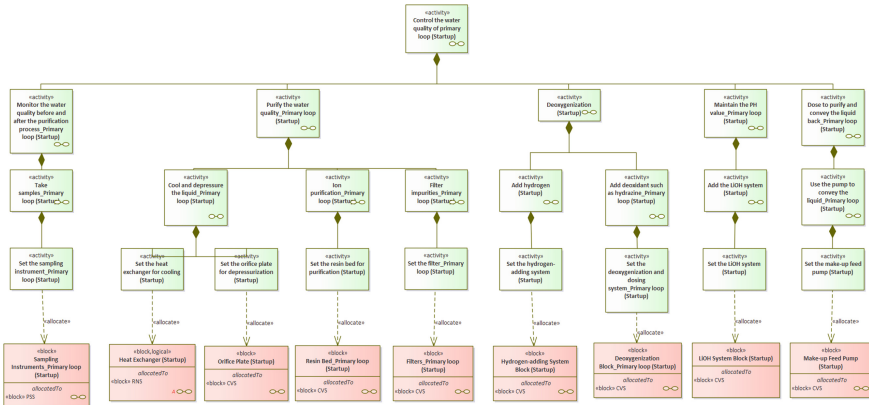


Fig. 5. Functional decomposition diagram of controlling the water quality of primary loop

### 5.2 Definition of Abnormal Conditions

To elicit abnormal operating conditions and define the failure mode based on the logical component structure, Fig. 8 defines an example of a “leakage” fault association for a surge line logic component (a surge line is a logical component that must function in the normal operating condition of a heating reactor). To cope with this failure, new design requirements need to be created to handle leakages.

Based on the new design requirements shown in Fig. 9, the heating reactor should have the function of handling pipeline leakage accidents, and the corresponding accident condition of the heating reactor is “leakage within the primary loop”. Accordingly, a system function “leakage within the primary loop” is defined based on this operating condition, and the function is functionally decomposed, and the underlying leaf node

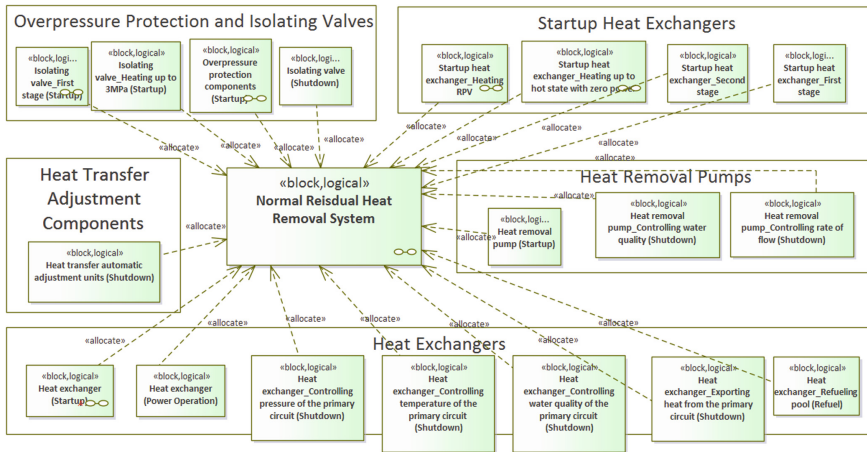


Fig. 6. Initial logical structure of normal residual heat removal system (normal operating condition)

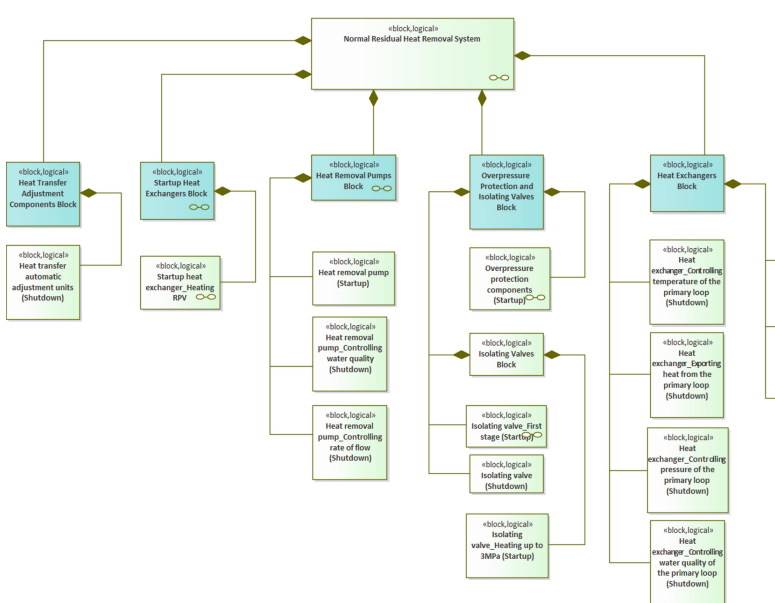
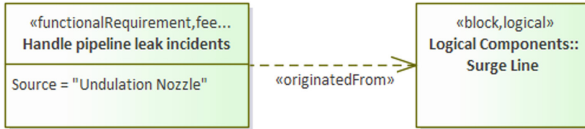


Fig. 7. Logical module composition definition of normal residual heat removal system (normal operating condition)

functions have logical structures corresponding to the implementation, and the logical architecture of the heating reactor system will be designed incrementally on the basis of these logical structures. Figure 10 shows the iterative logical structure of the normal residual heat removal system. Compared with Fig. 6, the new components are all used to alleviate the leakage condition of the primary loop.



**Fig. 8.** Example of a failure mode definition for a surge line



**Fig. 9.** New design requirements based on fault iteration

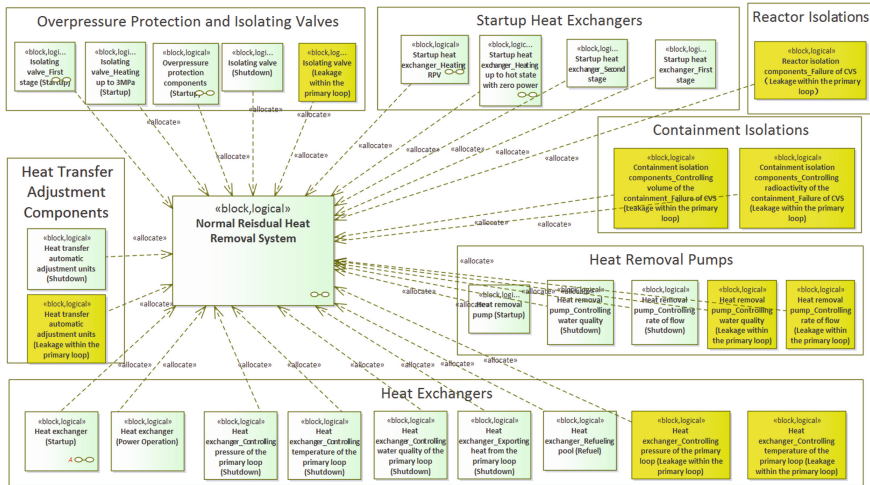
### 5.3 Internal Associations of the System Logical Architecture

After the logical structure of the heating reactor system is stable, the internal relationship of the system structure is defined. For the normal residual heat removal system of the heating reactor, its logical structure is repeatedly iterated, and the resulting logic module composition is: overpressure protection module(block), reactor isolation module, containment isolation module, heat removal pump module, heat exchanger module, startup heat exchanger module and heat transfer adjustment component module, the association between these modules will be defined according to the operating conditions, taking the start-up operating conditions as an example, the overpressure protection module of the normal residual heat removal system performs the relevant functions under the start-up conditions. In the same way, it can be inferred that the heat removal pump module and the start-up heat exchanger module are also involved in the system function implementation of the start-up condition, so the internal association of the relevant module under the start-up condition is defined. When all operating conditions are traversed, the complete logical architecture of the normal residual heat removal system is obtained, as shown in Fig. 11.

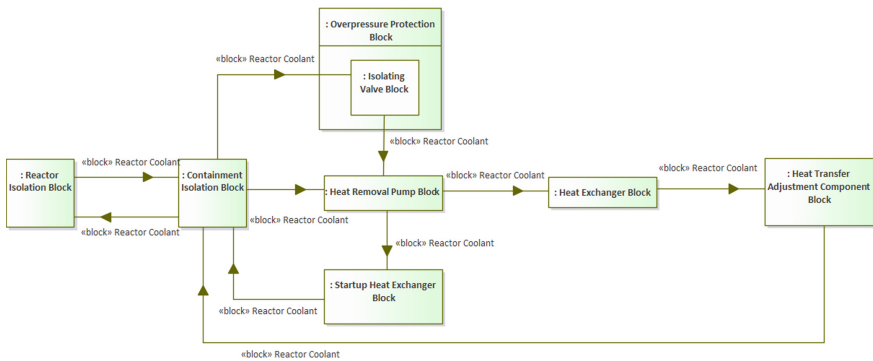
## 6 Summary

This study discusses how to deduce the analytical process and method of the functional and logical architecture of nuclear power system from the normal operating conditions. The main work is summarized as follows:

- 1) The incremental iterative generation method of nuclear power system architecture initiated by operating conditions is proposed: according to the process, when the semantic information related to the function and logical structure is complete, in addition to the functional decomposition link and the internal association definition of the logical architecture, the other generation processes can realize the process automatically.



**Fig. 10.** Iterative logic structure example of normal residual heat removal system (abnormal conditions)



**Fig. 11.** Example of normal residual heat removal system module internal association

- 2) The formal definition of the data structure of function and logic is given: based on the five-tuple structure, the formal definition of function and logical structure is described, as well as the relevant semantic constraint rules, which can further verify the semantic legitimacy of function and logic, and provide an inference basis for the automatic generation of logical architecture.
- 3) This paper proposes a design iteration based on logical structure faults: by including fault mode information in the logical structure, feedback design needs are induced, the abnormal operating conditions of the nuclear power system are derived, and the incremental progression of the top-level design of the system from normal operating conditions to abnormal operating conditions is realized.

This study is performed layer by layer according to the system engineering methodology, which is in line with the design habits of nuclear power systems. In addition, on the basis of the fault information of the logical structure, the user is supported to extend the definition to further support the system reliability analysis and other work. The example of heating reactor verifies the feasibility of this method for engineering practice in the existing MBSE modeling platform, and can provide guidance and reference for MBSE in the engineering of nuclear power system.

## References

1. Gang, X., Feiyue, W., Jiachen, H., et al.: Parallel system method for improving safety and reliability of nuclear power plant. *Syst. Eng.-Theory Pract.* **32**(5), 1018–1026 (2012)
2. Guocai, C., Feng, L., Huapeng, T., et al.: Design study of HPR1000 emergency reactor core waste heat discharge system based on system engineering method. *Nucl. Power Eng.* **42**(2), 6 (2021)
3. Peigen, Z.: Design of seismic automatic reactor stopping system based on systems engineering theory. *Autom. Instrum.* **38**(1), 42–45 (2017)
4. Chen, Y., Jianming, H., Xia, Y., et al.: Analysis of systems engineering methods applied to nuclear power design. *Inf. Syst. Eng.* **6**, 137–139 (2021)
5. Da, L., Zheng, W., Qian, W.: Application of systems engineering thinking in nuclear engineering. *China Nucl. Ind.* **9**, 44–45 (2020)
6. Lilin, L.: Practical analysis of systems engineering applications. *China Nucl. Ind.* **5**, 41 (2021)
7. Kun, P., Wenqiang, Y., Zhen, H., et al.: Design and application of model-based manned lunar spacecraft system. *Manned Spaceflight* **28**(1), 9 (2022)
8. ISO/IEC/IEEE 24748-4: Systems and software engineering—life cycle management—part 4: Systems engineering planning. ISO/IEC/IEEE 24748-4:2016 (2016)
9. Lin, Y., Yusheng, L.: Functional modeling method for conceptual design based on SysML. *Comput. Integr. Manuf. Syst.* **24**(1), 14 (2018)
10. Beitz, W., Pahl, G., et al.: *Engineering Design: a Systematic Approach*, 3rd edn. Springer, London (2007)
11. ISO/IEC 15288:2008 (IEEE Std 15288-2008). Systems and software engineering—system Life cycle processes.4.5[S], ISO/IEC 15288:2008