Modeling Method of Aircraft Simulation Based on Formal Description of Petri Net

Tianming Zhan[1], Yu Huang[2], Huapin Geng[3,4], and Jiahui Tong[3,4](✉)

[1] TAIJI Computer Co., Ltd, Beijing 100102, China
zhantianming@mail.taiji.com.cn

[2] Peking University, Beijing 100871, China

[3] Science and Technology on Complex System Control and Intelligent Agent Cooperation Laboratory, Beijing 100190, China
tjh80825@126.com

[4] Beijing Electro-Mechanical Engineering Institute, Beijing 102299, China

Abstract. The simulation of aircraft guidance and control system has characteristics such as multiple model classification, large parameter influence,complex information interaction and so on. The traditional code-level model development method has many limitations, such as time consuming, low reusability, low modeling efficiency, and model quality can not be guaranteed. Based on the concurrency, graphic representation, mathematics and simulation characteristics of Petri net, this paper studies the aircraft simulation modeling method of formal description based on Petri net, constructs a general modeling development and integration software tool that can carry out component, parametric and graphical modeling, solves the problems of multi granularity model modeling and automatic generation of real-time simulation model code, and realizes the visual graphic modeling and comprehensive integration of different granularity models such as components, modules and meta models. The research method has been successfully applied in the simulation modeling of an aircraft.

Keywords: Petri net · Modeling · Modeling software · Aircraft · Simulation

1 Introduction

The simulation of aircraft guidance and control system involves six degrees of freedom kinematics/dynamics model, inertial navigation system, guidance system, rudder system, thrust system, missile-target relative movement, environmental model calculation, etc. It has the characteristics of multiple model classification, large parameter influence, complex information interaction, etc. The traditional code-level model development method has limitations such as time-consuming and low reusability, leading to low standardization and generalization of models, low modeling efficiency, and model quality is not guaranteed, which

Y. Jia et al. (Eds.): CISC 2022, LNEE 950, pp. 863–871, 2022.
https://doi.org/10.1007/978-981-19-6203-5_86

is not conducive to the interoperation of simulation models and integration of simulation systems.

In order to solve the current problems of flight vehicle simulation modeling work, this paper proposes a vehicle simulation modeling method based on Petri net formalized definition, realized modular, parametric, graphical, and quickly simulation modeling for aircraft, and a simulation modeling tool was designed and implemented, which support users to build simulation model by graphical means, simulation calculation of the model.

2 Formal Simulation Model Based on Petri Net Description

In the process of aircraft simulation modeling, the following three aspects are mainly considered:

Hierarchical: The aircraft entity is composed of unit models, which are composed of component models, and component models are composed of metamodels;

Interaction Characteristics: There are close interaction characteristics among units, components and metamodels. The state update of each component is based on the state change of other parts, and the state change of each component will directly affect the running results of other parts;

Operation Characteristics: The models run at the same time to complete closed-loop iterative operation.

Since the simulation model has the characteristics of layers, interaction and operation, we made formal description of the aircraft simulation model by referring to the related knowledge of S Net extended from Petri Net and taking advantage of its characteristics of supporting time and data simulation.

2.1 Formal Definition for S Net [1]

Here, S Net's formal definition is given.

Definition 1: S net is a sextuple $SN = (P, V, T; F, R, Wr)$ where:

$$P \cup V \cup T \neq \emptyset$$
$$P \cap V = \emptyset \wedge P \cap T = \emptyset \wedge V \cap T = \emptyset$$
$$F \subseteq P \times T \cup T \times P \wedge R, Wr \subseteq V \times T$$
$$dom(F) \cup cod(F) = F \cup T \wedge dom(R \cup Wr) = V \wedge cod(R \cup Wr) \subseteq T$$

P is a set of places in SN, and it is used to represent the control states of the nets; V is a set of variables and it is used to represent all variables in nets; T is a set of transitions; F is a set of arcs (or flow relations); R is a set of read relations and Wr is a set of write relations. R and Wr are used to read and write data by transitions, respectively [2].

dom and *cod* are defined as follows:

$$dom(r) = d1 | \exists d2 \in D2 : (d1, d2) \in r$$

$$cod(r) = d2 | \exists d1 \in D1 : (d1, d2) \in r$$

And then there are some basic operations defined on S net.

Definition 2: Basic operations

Given x and t, $x \in V, t \in T, p \in P$:

$\bullet t, t\bullet, \bullet p, p\bullet$ are still pre set and post set in directed net;
$r(x) = t | (x, t) \in R$ is called read set of x;
$w(x) = t | (x, t) \in Wr$ is called write set of x;
$r(t) = x | (x, t) \in R$ is called read set of t;
$w(t) = x | (x, t) \in Wr$ is called write set of t;
$tp(x)$ is used to represent the type of x;
$Type = \bigcup_{x \in V} tp(x)$is called type set of SN.

Definition 3: $\sum = (P, V, T; F, R, Wr, K, W, MT, MO, T, Clock)$ is ST net system where:

$SN = (P, V, T; F, R, Wr,)$ is S net;
$K : P \rightarrow \{1, 2, ...\}$,is called capacity function;
$W : F \rightarrow \{1, 2, ...\}$,is called weight function;
MT is called transition mark of SN;
$M0$ is the initial mark;
T is the time mark;
$Clock$ is the system clock [3].

2.2 Simulation Model Description Language Based on S Net [4]

We use the extended description language $ST\ PNML$ to transform S net system model into a form that can be understood by computer, including place, variable elements, transition, read/write relationship, time identification and so on.

The XML rules mainly include Reuse Elements(including names, streams, values, capacities, time identifiers, etc.), Library, Transition(including Transition condition, implementation body, DLL path, Transition), Variables, Connection Arc.

3 Modular Simulation Modeling Method Based on Petri Net Description

3.1 System Framework

The simulation modeling software is divided into three levels: GUI layer, kernel layer and database layer.

GUI Layer: Provides user interfaces, such as project display interface, visual modeling interface, model display interface, information configuration interface, template library management interface, code generation interface and so on.

Kernel Layer: Is the kernel part of the whole simulation modeling software, including all subsystems. There are six parts: project management subsystem, modeling subsystem, template management subsystem, data display subsystem, code generation subsystem and software integration subsystem.

Database Layer: Stores model data files, project data files, and code data files, provides standard descriptions of data structures such as model design, module grouping, mathematical algorithms, data export, and code export.

The system framework is shown in the Fig.1.

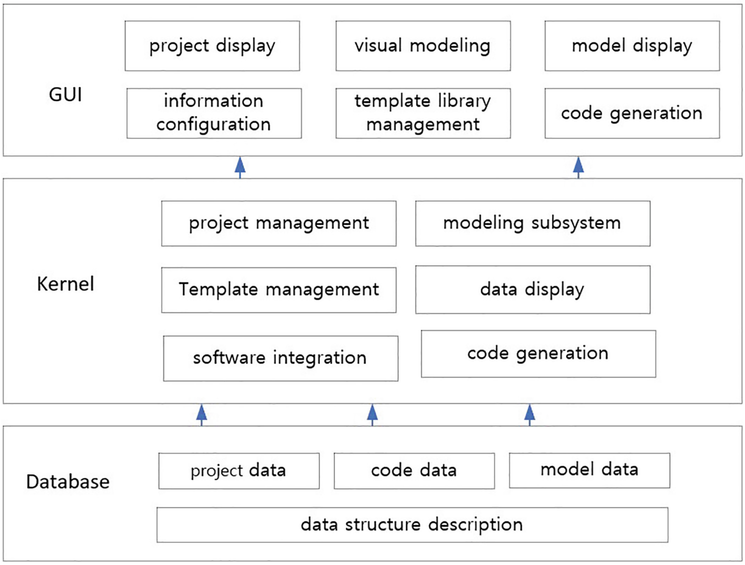

Fig. 1. System architecture diagram

3.2 System Composition

The general simulation modeling software mainly includes six subsystems: engineering management, simulation modeling, template library management, data presentation, code generation and software integration subsystem, which are respectively introduced as follows:

Engineering Management Subsystem: Manages all aircraft engineering projects, implements project creation, modification, deletion, directory management, and simulation information management of models, components, variables, results and data, which is the entrance of other subsystems;

Simulation Model Modeling Subsystem: Including component visual rendering, model configuration, variable mapping configuration, initialization configuration and other functions;

Template Management Subsystem: Manages template resources for simulation modeling. Including component and component template classification management, template design (variable design, algorithm design, interpolation algorithm binding, simulation step size, template information), algorithm library management;

Software Integration Subsystem: Integrate the software and ensure that the modeling results are consistent with the exported code;

Code Generation Subsystem: Generates code for a specific purpose and supports synchronization of changes to code files into modeling software. Including component and component code generation, and simulation process control code generation;

Software Framework: Software support framework, interfaces between subsystems, databases and operating systems.

3.3 Subsystem Structure Relationship

The relationship of each subsystem is shown in the Fig. 2.

The aircraft simulation model was decomposed according to the principle of subsystem correspondence, and the Unit level model in the model architecture was obtained. The Unit level model was decomposed according to the principle of minimum independent logic algorithm, and the component level model was obtained. Through this series of decomposition, the modular work of the simulation model is completed. The general model is formed by extracting the parameters of various models. By combining the Unit level models and assigning values to the extracted parameters, the Unit level model modeling is completed. The aircraft model can be formed by combining multiple Unit level models.

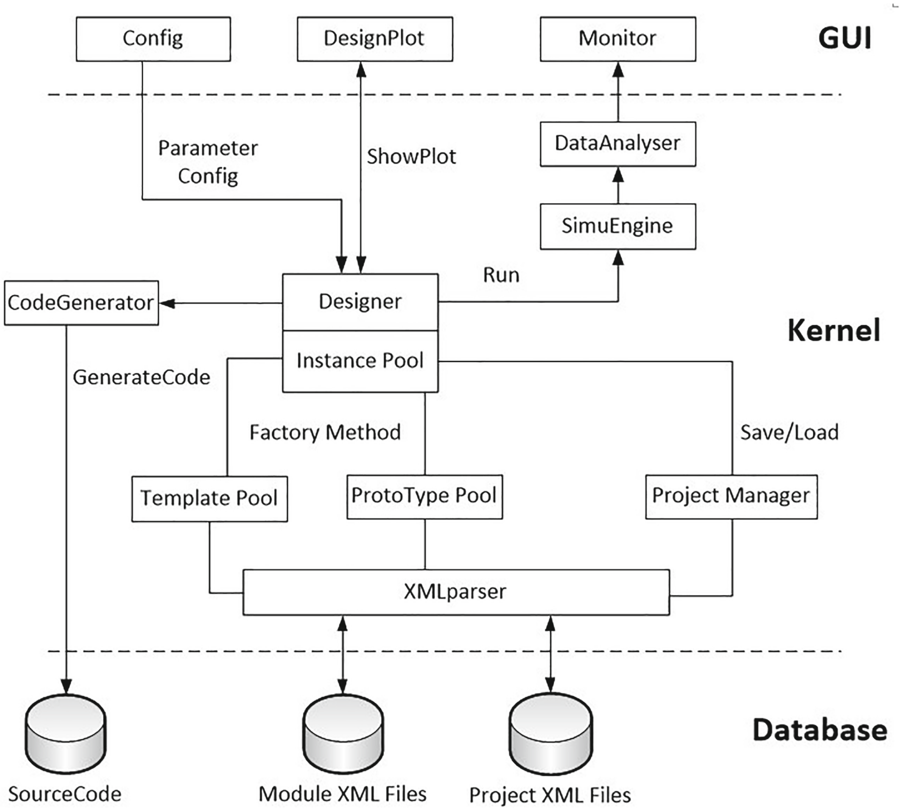

Fig. 2. Structural relationship of each subsystem of the platform

4 Modeling Process

4.1 Simulation Modeling Process Design

The minimum model structure in the general modeling platform is the meta-model, which includes model algorithm, initialization interface, I/O interface, model relation interface, etc., as shown in Fig. 3.

The simulation modeling process is shown in Fig. 4.

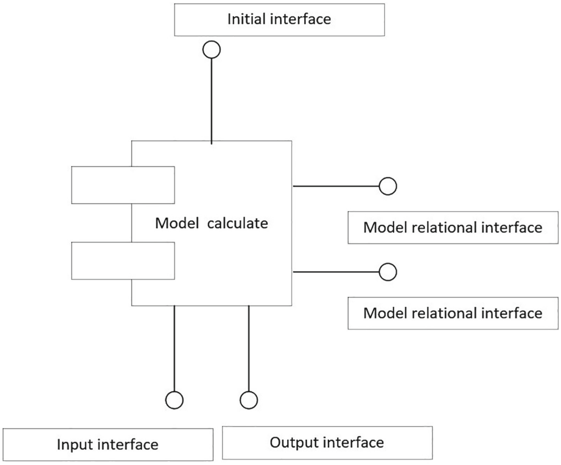

Fig. 3. Schematic composition of a metamodel

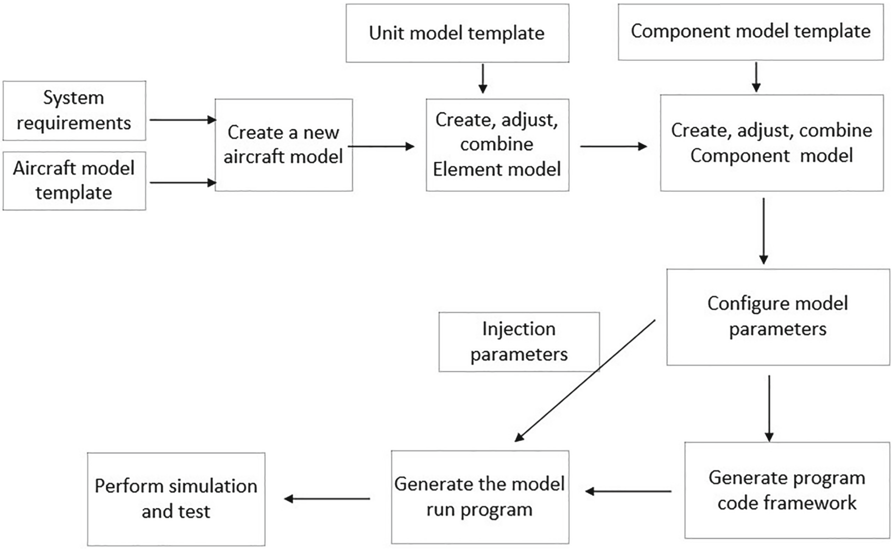
Unit model template
Component model template
System requirements
Aircraft model template
Create a new aircraft model
Create, adjust, combine Element model
Create, adjust, combine Component model
Configure model parameters
Injection parameters
Perform simulation and test
Generate the model run program
Generate program code framework

Fig. 4. Schematic diagram of simulation modeling process

4.2 Design of Simulation Model Libraries

Modeling software includes three types of libraries:

Component Model Libraries: Is a collection of basic model components, which perform some specific functions and describe simulation objects from different characteristics.

Unit Model Library: Is a collection of instantiated models of aircraft functional units. A unit model can be generated by combining several component models.

Aircraft Model Library: Is a collection of aircraft composed of parameterized unit models.

4.3 Code Generation

Based on the graphical modeling results, the modeling software generates a model project according to certain rules. Each component unit generates a C++ class, including a *.h* file and a *.cpp* file, in which the corresponding variables and function names are defined. The generated C++ code includes relationships between components and modules, as well as transitive relationships between variables.

5 Examples of Application

Based on the graphical modeling and simulation platform software, a simulation model of an aircraft is established and simulation analysis is carried out. The simulation results show that the modeling software can quickly build the simulation model, set the data flow relation, I/O parameter mapping, and automatically generate the code framework. It can realize effective management of aircraft model and general algorithm, and monitor simulation results in real time (Table 1).

Table 1. Comparison of modeling methods

Performance indicators	Traditional C	This paper	SIMULINK
Modeling approach	Edit code	Graphics operation	Graphics operation
Average modeling time	30 days	13 days	21 days
Readability of code	Strong	Strong	Weak
Model reuse	Weak	Strong	Weak

Compared with the Traditional C++ code modeling:

- Intuitive, convenient to operate in the modeling process;
- Configure quickly interactions between models;
- Provide a standard model library to reduce repetitive coding;
- More standardize about the description and implementation of the model.

Compared with SIMULINK:

- The model can be reused so that users can avoid building the model layer by layer from the lowest functional unit;
- The generated model code is consistent with the model architecture, and the code is highly readable;
- The model supports realtime simulation.

6 Conclusion

This paper studies the aircraft simulation modeling method based on Petri net formal description, Developed componentized, parameterized and graphical general modeling and simulation software; By using drag and drop modeling method, aircraft simulation components are built quickly, Established data flow relation and input and output parameter mapping; Realized efficient management of model components, solution algorithm library and data file library, The problems of multigranularity model integration and realtime simulation model code generation are solved; The visualized graphics modeling and comprehensive integration of different granularity models such as component, module and metamodel are realized, which is successfully applied in a certain aircraft simulation modeling.

References

1. Yuan, C.: Principle and Application of Petri Net. Electronic Industry Press, Beijing (2005)
2. He, Y., Huang, S., Liu, M., Huang, Y.: Process model of collaborative design based on petri nets. J. Syst. Simul. **S1**, 11–15 (2011)
3. Tang, S., Yu, L.: Research on missile confrontation simulation based on reference petri net. Flight Dyn. **31**(1), 88–92 (2013)
4. Zhou, F., Wu, B., Gu, Q.: Workflow modeling and correctness analysis based on petri net. Comput. Sci. **32**(2), 121–124 (2005)