Satellite Dynamic Fault Analysis Method Based on AltaRica Logic Modeling

H Wang, Y Li, Z Wang and J Wu
Institute of Telecommunication Satellite, China Academy of Space Technology, Beijing 100094, China
E-mail: wanghongxun1990@163.com

Abstract. Modern spacecraft has developed into a complex system integrated multi-task, multi-specialty and multi-information. The cross-linking and coupling between different systems have exposed more and more logic sequence fault caused by non-software and hardware failures. It is difficult to comprehensively and accurately analyze the logical sequence faults of complex systems by using traditional fault tree analysis (FTA), failure mode and effect analysis (FMEA) and other manual methods. Furthermore, the reliability and safety of the spacecraft cannot be effectively guaranteed. In this paper, the dynamic fault analysis of the spacecraft was carried out based on the method of AltaRica logic modeling. The satellite power supply and distribution (PSD) subsystem was taken as an example. A dynamic model of charge and discharge between the solar array, the storage battery and the bus was established. Logical events and sequences of the bus power supply fault was calculated. The results show that this method can effectively complete the dynamic modeling and fault analysis of the spacecraft system, and it has higher analysis capability and efficiency than traditional methods.

1. Introduction
Due to the high reliability and long life of the spacecraft, the on-orbit fault tolerance capability design is vigorously promoted during the development of the spacecraft. The fault analysis technology as the basis of the fault tolerance design has played an increasingly important supporting role in the identification and control of the potential risk of the spacecraft. However, modern spacecraft has developed into a complex system with multiple systems, multiple environments, multiple missions and multiple resources that are interconnected, supported, integrated, and constrained. The spacecraft function has the technical characteristics of multi-object, multi-information, multi-specialty, multi-task, multi-resource and multi-process integration, which brings huge challenges to fault analysis. Traditional fault tree analysis (FTA), failure mode and effect analysis (FMEA), etc., which mainly rely on manual analysis methods, are difficult to achieve accurate analysis of such complex systems. The huge workload of fault analysis of complex systems leads to low efficiency, which will result in the fault analysis cannot be synchronization with the system development process.

In addition, due to the complex cross-linking and high coupling between different systems of the spacecraft, many undesirable events caused by non-software and hardware failures are exposed in modern spacecraft systems. These undesirable events are caused by logical sequence errors between different events in the system. Such as, the logic sequence of charge and discharge between the solar array and the storage battery in the satellite power supply and distribution (PSD) subsystem, The
potential timing failure of the input and output ports of the output chip of the satellite service management unit due to the use of different power supplies, the sequence failure of the central remote control circuit due to the energization sequence, the timing failure caused by power-on delays in the switching circuit of the secret switch. Traditional FMEA and FTA methods cannot characterize such logical relationships and sequence effects, so a dynamic model needs to be established for analysis. However, there is no effective dynamic analysis technology to carry out system analysis during the development of spacecraft. The dynamic fault analysis technology based on AltaRica logic modeling provides an effective method for solving such problems.

AltaRica is a high-level formal description language specifically for complex systems, which is developed by La at the University of Bordeaux and his industrial partners. The fault analysis technology based on AltaRica logic modeling is to use the system's constituent units, unit status, logical relationships within and between units and fault data to conduct unified modeling, and then to conduct reliability and safety analysis based on the unique model built. The static model established by using AltaRica language can automatically generate the results required by the reliability safety design of the system, such as fault tree, reliability block diagram, FMEA, etc., without the need to model the reliability and fault tree separately like traditional analysis methods [1-4]. At the same time, dynamic modeling of complex systems is achieved by establishing the timing and logic transition information between different units in the model. The technology has been widely used in aviation, space technology, defense, energy, transportation, maritime and other fields.

In the aviation field, fault analysis technology based on AltaRica logic modeling has become a mature and indispensable technology for reliability and safety analysis of foreign aviation systems, and is widely used in all aspects of the aviation field, including overall design, products supply, inspection and testing, maintenance and so on, among which the Centre National d’Etudes Spatiales and Airbus are the most widely used [5]. In China, the model-based fault analysis technology has become an important technical means for many important scientific research institutes, universities and engineering application to carry out reliability and safety research and application. Researchers [6-8] from these organizations conducted the comparative study of the fault analysis method and the traditional fault analysis method in the field of aviation. The results show that the AltaRica based method has many advantages in the modeling of aviation systems. In the aerospace field, the fault analysis technology based on AltaRica logic modeling is increasingly recognized abroad, and its application is becoming more and more extensive. Only in the past 20 years, major spacecraft projects such as Hermes, Ariane Rocket, International Space Station, European Automated Cargo Ship, Galileo Satellite Positioning System, etc. have adopted this technology to conduct system reliability and safety analysis. However, the application of this technology in the domestic aerospace field is still to be studied and verified.

In this paper, the application of dynamic fault analysis technology based on AltaRica logic modeling in spacecraft development is carried out. Firstly, the modeling and analyzing method for dynamic faults is introduced. Secondly, the satellite PSD subsystem is used as a case for safety sequence modeling, and the dynamic fault analysis is performed based on the built model. Finally, the conclusions and future work of this study are discussed.

2. Modeling methods
Fault analysis based on AltaRica logical modeling uses the principle of constraint automata. To build a complete constraint automata containing the following main information: unit block b, input i, output o, state variable s, logical Function f, as shown in Figure 1. The basic unit of the model is the unit block. Each unit block represents a sub-system, module, single machine or component. Each unit block is one or more constraint automata of different levels and complexity.

The output of the constraint automata establishes the logical relationship between the input information and the internal state variables through the defined specific logical function as shown in equation (1):
\[ o = f(i,s) \]  \hspace{1cm} (1)

Input and output interfaces are channels for information transition between different blocks. Each input or output can have one or more states at the same time as shown in equation (2), where \([o_1, o_2, \cdots, o_n]\) means different states of output. The interface variable is a parameter that reflects the state of the input and output information of the block. Any number of input or output states can be defined as needed. The state of \([o_1, o_2, \cdots, o_n]\) can be “nominal” or “failed” and “degraded” three types.

\[ o = [o_1, o_2, \cdots, o_n] \]  \hspace{1cm} (2)

The state variable is a parameter that indicates the unit block’s own hardware and functional status during the life cycle as shown in equation (3). \(N\) in the formula represents the nominal state of the unit block. \([F_1, F_2, \cdots, F_n]\) represents all possible failure modes, and it can be failure and degradation. The state variables inside the constraint automata can be changed or transitioned when triggered by external events as shown in equation (4). While defining the state variables of the unit block, the type and corresponding parameters of each fault probability density function are set according to the fault of the components characterized by the unit block. All faults in this research are exponential.

\[ s_n = [N, F_1, F_2, \cdots, F_n] \]  \hspace{1cm} (3)

\[ s = \begin{cases} s_0, & \text{if } (e = e_0) \\ s_k, & \text{if } (e = e_k, k = 1, 2, 3, \cdots) \end{cases} \]  \hspace{1cm} (4)

The logical function \(f\) of the unit block is an equation that affects the relationship between the input and state variables of the unit block on the output result, and it is the key to establish the relationship between the output of the unit block and the input and fault modes. The output state of the unit block is determined by each input state and the unit block state. The definition of the logical function \(f\) uses the method of AltaRica script and logical polynomial to determine the specific impact of the input information and state variables in the unit block on the output. The logical relationship between different unit blocks is established by information transition between the unit blocks. The output of all the unit blocks before the unit block \(b_{m+1}\) may be the input of the subsequent unit block as described in equation (5), where the input of the unit block \(b_{m+1}\) is determined by the logic function \(g\). Through a series of complex logical functions between constraint automatas, the logical relationship between the output of the entire model and each unit block is established.

\[ i_{b_{m+1}} = g(o_{b_1}, o_{b_2}, \cdots, o_{b_m}), m=1, 2, 3, \cdots \]  \hspace{1cm} (4)
State transition refers to the phenomenon that the function of a system, machine or component itself changes from one state to another when an event occurs. Some systems, machines or components have different functions at different stages or under different conditions in the service process, and the transition between different functions requires certain conditions, such as the solar array and storage battery in the satellite PSD subsystem. In the case of sufficient light, the solar array supplies power to the bus and charges the storage battery. When there is no light, the solar array stops supplying power, and the storage battery supplies power to the bus.

A system with state transitions has a sequence in time relationship between the occurrence of different events. The static model cannot characterize the impact of time sequence changes on the system, so a dynamic model needs to be established. The definition of state transition is only used in dynamic models, and the process of state transition is determined by defining the logical relationship between the external triggering event, the subject of the state transition and the influence of the transition.

The logic modeling process is mainly contains five steps as shown in Figure 2. First, system logic analysis of the modeled objects is required. Determine the research object (system) according to the requirements, and decompose the function and hardware composition of the research system step by step, until the bottom unit of the built model, such as a machine or component, is determined, and then the failure mode of the bottom unit itself is determined. Second, the modeling and logic definition of the unit block from top to bottom based on the system logic analysis is carried out. In this study, the modeling of the unit block adopts a top-down approach (consistent with the system design method), that is, the top-level unit of the system decomposes the functional modules level by level, and then adds input and output interfaces to the built unit block, and defines the interface attributes, State and the state of this layer unit block itself. Third, a static logical relationship between the unit block input, state and output is established. This step can be achieved by polynomial method and AltaRica language. The fourth is establish the dynamic logical relationship of the unit block. Set event attributes according to the system mission profile, and establish dynamic logic by setting external trigger events and state transition functions, and define the initial phase of the system. The fifth is model verification and calculation. According to the differences in modeling requirements, the modeling process and output results are also different. For the established model, a step-by-step simulation method is used for verification, so as to improve the model.

3. Case research of dynamic modeling
3.1. System analysis and modeling ideas
The satellite PSD subsystem uses the solar array as the main power source, and the storage battery as the energy storage device. The power conditioning unit (PCU) regulates and controls the power supply bus. In the light area, the solar array provides stable and sufficient power to the star. During the no or weakening light area, the output capacity of the solar array is insufficient, the storage battery will provide the whole star with combined power supply with the solar battery or independent power supply by the storage battery to ensure the power demand of all electric loads of the whole star. After the power is restored to meet the power requirements of all the electrical loads of the whole star, the extra output power is used to charge the storage battery, so that the state of charge of the storage battery is always maintained in the safe range of power supply that can guarantee the normal and unexpected energy of the whole star.

The dynamic loop between the solar array, storage battery and PCU is the main loop of the entire PSD subsystem. In this loop, there is a dynamic sequential logic of alternating power supply from the solar array and the storage battery. First, the top-level modeling architecture design is carried out according to the functional principle of the PSD subsystem to realize the structural framework and basic functions. Then, the module status, information flow, failure mode, failure rate, state transition and other information in the model are defined in detail. Finally, through setting the initial working conditions of the system to simulate the operation of the system for calculation and analysis.

3.2. Model architecture
The PSD subsystem is composed of a solar array, a storage battery, a bus, and a Power conditioning unit (PCU) composed of battery charge management (BCM) and main error amplifier (MEA). The satellite is powered under the influence of periodic light. The system architecture is shown in Figure 3:

![System architecture diagram](image)

**Figure 3.** The architecture of PSD subsystem model.

The satellite PSD subsystem includes the following main parts:
1) External environment: light
2) Power supply: solar array, storage battery
3) Power supply control:
   - MEA: power supply selection
   - Bus: power input / output
   - BCM: charging control

When the satellite runs to the lighted position, the solar array starts to output power after receiving the light. At this time, the MEA selects the solar array to supply power to the bus and the battery does not supply power. After receiving the power from the solar array, the bus provides power to the entire satellite system. At the same time, the solar array also starts to provide the charging output for the storage battery. The charging output charges the battery after BCM charging control.
When the satellite runs to the position without light, the voltage of the solar array decreases. At this time, the MEA selects the storage battery to supply power to the bus, and the bus continues to supply power to the entire satellite system. Meanwhile, the bus no longer provides the charging output for the battery, and the BCM charging control stops charging the battery.

In this way, the PSD subsystem can continuously provide power for the normal operation of the satellite.

### 3.3. Block definition

The definition of unit blocks such as light, solar array, storage battery, MEA, BCM and so on involved in the model is as follows:

1) **External environment: light**
   - The states of light are defined as:
     - right
     - wrong
   - The state of light is simplified to “light” and “no light” without considering the intermediate state.
   - The change of light state is simplified as a random trigger event in the model.

2) **Power supply:** solar array, storage battery
   - **Solar array**
     - The states of the solar array are defined in the model as:
       - normal power supply (nominal)
       - insufficient voltage (degraded)
       - failure
     - The transition between “normal power supply” and “insufficient voltage” is triggered by light conditions. The “failure” is a random event and cannot be repaired. The constraint automata principle of the solar array as shown in Figure 4.
   - **Storage battery**
     - The storage battery states are defined as:
       - normal power supply (nominal)
       - insufficient voltage (degraded)
       - failure
     - The transition between the two states of “normal power supply” and “insufficient voltage” is triggered by the BCM charge control output. The “failure” is also a random failure and irreparable. Random failure of the solar array will eventually lead to insufficient storage battery voltage.

![Figure 4. The architecture of PSD subsystem model.](image-url)
(3) Power supply control: MEA, bus, BCM

1) MEA: power supply selection

MEA states are defined as:
- nominal
- failure

The MEA can randomly fail and cannot be repaired. When the MEA fails or neither the solar array nor the storage battery can supply power normally, the MEA power supply is selected as “no power supply”; when the MEA is normal, the power supply is selected as which can normally supply power.

2) Bus: power input / output

The states of the bus are defined as:
- nominal
- failed

The bus itself can randomly fail and cannot be repaired. When the bus itself fails or the MEA power supply is selected as “no power”, the bus has no power output, that is “failed”.

3) BCM: charging control

The BCM states are defined as:
- nominal
- failed

The BCM itself can randomly fail and cannot be repaired. When the BCM fails or the bus is not powered, the BCM charging control is not charged.

After the above analysis and design on satellite PSD subsystem, the model is established as shown in Figure 5:

![Figure 5: The established model of PSD subsystem.](image)

3.4. Calculation and results

In the dynamic model, we define the “bus power supply = failed” is defined as the “undesirable event". The initial state of the system is defined as: "The satellite is currently operating in a position without light, the voltage of the solar array is insufficient, and the storage battery is supplied with power, and all components are normal and without failure. The maximum sequence order of undesired event is set to three, and all event sequences that can cause undesired events can be calculated, and the corresponding event sequence can be calculated as shown in Table 1."
The calculation identified five first-order sequence events that caused the bus power supply to fail, including “MEA failure”, “solar array failure”, “BCM failure/storage battery voltage shortage”, “storage battery failure” and “bus failure”; and one third-order sequence events, that is “Light / Solar array normal power supply / Storage battery insufficient voltage / BCM failure / No light / Solar array insufficient voltage”.

Table 1. Logical events and sequences calculated.

| Number | Order | Spacing |
|--------|-------|---------|
| 1      | 1     | MEA failure |
| 2      | 1     | Solar array failure |
| 3      | 1     | BCM failure / Storage battery insufficient voltage |
| 4      | 1     | Storage battery failure |
| 5      | 1     | Bus failure |
| 6      | 3     | Light / Solar array normal power supply / Storage battery insufficient voltage / BCM failure / No light / Solar array insufficient voltage |

The first-order sequence event means that only this event will cause the failure of the bus power supply. For the BCM failure/Storage battery insufficient voltage in the first-order sequence, BCM failure is the direct cause of the failure of the bus power supply, and storage battery insufficient voltage is intermediate process result after BCM failure occurs.

The third-order sequence means that the three events must occur in specific sequence that can lead to the bus power supply failed. The analysis of the six events in this third-order sequence is as follows. The initial working condition of the system is no light. When there is light, the system switches to the solar array for power supply. The “Solar array normal power supply” and “Storage battery insufficient voltage” are the results of the intermediate process after the light occurs. Then the event “BCM failure” occurs, and the storage battery will not be charged continually at this time, but the solar array continues to supply power. if the event “No light” continues to occur, the bus power supply of the entire system will fail, and the “Solar array insufficient voltage” is an intermediate result after the event “No light” occurs. Therefore, the direct cause combination event for this third-order sequence is “Light / BCM failure / No light”. In other words, when the event “light”, “BCM failure”, “No light” occurs in sequence, it will cause the bus power supply to fail.

4. Conclusions and future work

In this paper, a dynamic fault analysis method based on logical modeling was carried out. The satellite PSD subsystem was taken for case study. A dynamic charge and discharge model between the storage battery, solar array, PCU and bus was established based on the dynamic logic relationship. And five first-order sequence events and one third-order sequence events that can cause the bus power supply to fail were identified.

The dynamic fault analysis method based on logical modeling can well realize the dynamic system modeling and analysis. And it has higher fault analysis capability and efficiency compared with traditional manual method. If the speed and time information of charge and discharge are defined in the model built in this paper, it is possible to further simulate the impact of storage battery performance degradation on the availability of satellite PSD subsystems. Later, in order to analyze the satellite system more comprehensively, a combination of static and dynamic models can be used. First, establish a static model of the system, and the dynamic events in the system are modeled separately. Then, embed the dynamic model into the static model for dynamic analysis based on the static fault analysis. The combination of dynamic and static methods can more realistically simulate
the complete logic of the entire system, and it can identify faults that are difficult to find with traditional methods.

Acknowledgments
This research is financially supported by the National Science and Technology Major Project Foundation of China (No. 2017ZX01013101-003).

References
[1] Antoine R and Chaire B F 2014 10th France-Japan/ 8th Europe-Asia Congress on Mecatronics (Tokyo: IEEE Xplore) 1-10
[2] Mortada H and Prosvirnova T 2014 Lect. Notes Comput. Sc. 8822(1) 181-94
[3] Long G and Liang A 2017 Product failure modeling method based on Altarica. language Prognostics and System Health Management Conf (Harbin: IEEE Xplore) 1-6.
[4] Li S and Li X 2014 Procedia Eng. 80 140-52
[5] Che C and Liu Y 2016 Adv. Aero. Sci. and Eng. 7(3) 369-73
[6] Xu W and Zhang Y 2017 Comput. Eng. and Sci. 39(12) 2269-77
[7] Gu Q, Wang G, Zhang L and Zhai M 2015 Comput. Sci. 42(3) 124-43
[8] Zhu Y, Zhang J G, Gong Q, Fan Y, Wang P and Wang C 2011 9th Inter Conf on Reliability, Maintainability and Safety (Guiyang: IEEE Xplore) 588-93.