

# Method of Key Rating Ascertainment for Space Software based on Fuzzy Synthetic Evaluation and Quantitative Fault Tree Analysis

Li Yang

Changchun Institute of Optics, Fine Mechanics and Physics,  
Chinese Academy of Sciences  
CIOMP  
Changchun, China  
victoria\_liy@163.com

Song Kefei

Changchun Institute of Optics, Fine Mechanics and Physics,  
Chinese Academy of Sciences  
CIOMP  
Changchun, China  
songkefei@sina.com

**Abstract**—Software key rating ascertainment is very important in development of space software. It can guide manager develop the congruent software using the optimal benefit-cost ratio in management. It can also provide the direction in software measurement, and help designer improve system reliability finally. In view of space software characteristics and quantitative data shortage, a method of key rating ascertainment for space software based on fuzzy synthetic evaluation and quantitative fault tree analysis is proposed. The results of application to space software development show that the new approach makes software key rating ascertainment have the quantificational evidence, which is ascertained using quantitative data and qualitative analysis.

**Keywords**—space software; key rating; fuzzy synthetic evaluation; quantitative fault tree analysis (QFTA); bottom event; top event

## I. INTRODUCTION

The scope of computer application is being applied more and more widely in the world. The software is maturing into the core of key system in aerospace and become increasingly complex. Sometimes only one control system consists of a dozen computer software configuration items. In order to develop the software which falls in with requirements in optimal proportion of expenditure to efficiency, software must be managed according to the key rating and its scale.

Usually severity and occurrence probability of dangerous should be considered. Controlling of hazardous by software, the complexity and real time of controlling also must be considered. Because of the prodigious subjectivity, how to quantificationally define the key rating of software exactly is an aporia when these factors are thought over.

The method based on vague set to quantificationally analyze fault tree, and thereby ascertain software key rating was put forward by Li Yang in 2010[1]. Data such as code line, interface relation and so on is insufficient when the key rating in application process is determined. So the ascertainment of these parameters relies on the experience of experts in this area who make decision fuzzy. Specialist grading can used reluctantly, but the measure cannot get the satisfactory outcome because of congenitally deficient. Accurate mathematics and probability theory are difficult to deal with fuzzy problem. Drawing fuzzy

theory into key rating ascertainment is the need of engineering practice. A Method of key rating ascertainment for space software based on fuzzy synthetic evaluation and quantitative fault tree analysis is proposed to solve the above-mentioned problems [2].

## II. THE KEY RATING ASCERTAINMENT OF SPACE SOFTWARE

The ascertainment of software key rating aims at software configure item, because space software is managed by configure items usually. So risk event and risk reason must be analyzed first of all when software key rating is fixed on. Secondly the grade of risk ponderance and the occurrence possibility of risk event are ascertained according to the occurrence relative probability of risk event. Sequentially the system risk target and the control sort of software are ascertained. At last, software key rating is calculated.

The method of fuzzy synthetic evaluation bases on fuzzy mathematics. It translates qualitative evaluation into quantitative evaluation according to the theory of membership grade in fuzzy mathematics. This method can evaluate object or phenomenon totally which is restricted by multifarious factors. The sixty-four-dollar question of the measure is how to calculate fuzzy evaluation matrix.

The foundation of the new method is setting up the fault tree to analyze risk fountain. The first step is calculating the relative synthetic evaluation of bottom event based on fuzzy synthetic evaluation. The software key rating is getting finally using the occurrence possibility of risk and risk coefficient.

### A. Risk fountain analyzing using fault tree

Fault tree is established taking configuration item invalidation as top event and component invalidation as bottom event. Analyze risk fountain using the relation of top event and bottom event.

### B. Modeling and relative probability calculation of bottom event based on fuzzy synthetic evaluation

The most pivotal step is analyzing the risk's reason, and making sure the grade of risk ponderance and the occurrence

relative possibility of risk event according to the occurrence probability of risk event in the process of whole software key rating determination. Because the software is very complex now, using qualitative technique to determine the rating is not satisfying. But the exact quantificational data can be achieved difficult. So the new method banding the qualitative analysis and quantificational calculation together is put forward.

#### 1) Assuring the set of synthetic evaluation factors

The set of evaluation factors is the collection of factors which affecting the happening of bottom events. It is often expressed as  $U = \{u_1, u_2, \dots, u_n\}$ . The bottom event factors who affect the ascertainment of space software key rating probability of bottom event are reduced to  $u_1$ : effective rows of code (The unit is thousand rows),  $u_2$ : the density of software error (The level is higher and the density of software error is lower),  $u_3$ : the weighted factor of means validity which is adopted to prevent the occurrence of bottom event (There is correlation between value and validity. Values are between 0 and 1. The higher the validity and the smaller the value),  $u_4$ : the software maturity weighted factor (Values are between 0 and 1, the higher the validity and the smaller the value. Generally the value of new software is 1.),  $u_5$ : the number of software developers (The association between the number of people and bottom event likelihood is: the more the person, the higher the likelihood),  $u_6$ : the number of inner interface (the more the interface, and the bigger the likelihood),  $u_7$ : the number of exterior interface (the more the interface, and the bigger the likelihood). So the set of synthetic evaluation is  $U = \{u_1, u_2, \dots, u_7\}$ .

#### 2) Ascertaining the weight of evaluation factor

The evaluation weight is relative order of importance of all factors which induce occurrence of bottom event.

It is expressed as  $A = \{a_1, a_2, \dots, a_n\}$  usually, where  $a_n \in [0, 1]$ , and  $\sum_{i=1}^n a_i = 1$ . The method of brainstorm is used to ascertain final weight in order to ensure the relative accurate of relative importance degree. The final weight value is average of each factor's weight. Each factor's weight is given by one software designer, one reliability designer and one designer who has prolific experience but does not take part in designing. They evaluate the weight based on condition of history data, development, and experience and so on. The value given by above experts is established order from big to little. The maximum one multiplies one, the middle one multiplies four, and the minimum multiplies one. The final result  $a_i$  is above sum of calculation divided by six. So the set of weight  $A = \{a_1, a_2, \dots, a_7\}$ , thereinto,  $a_n \in [0, 1]$  and  $\sum_{i=1}^n a_i = 1$ .

#### 3) Ascertaining remark set

Comment set is the circumstances description of each factor, expressed as  $Y = \{y_1, y_2, \dots, y_m\}$ . It is synthetic evaluation of each

factor that arouses the happening of bottom event. If the comment set is too many, standard would be predominated disadvantaged. But it is not beneficial to calculate when comment set is too few. At the same time, comment set must be applicable to all factors and the amount of comment set must be appropriate. The comment set is defined according to the characteristic of space software:

$Y = \{\text{more possible, very possible, possible, unlikely, impossible}\} = \{Y_1, Y_2, Y_3, Y_4, Y_5\}$ .

#### 4) Ascertaining evaluation matrix R

$R$  is fuzzy mapping of  $X \rightarrow Y$  which is used to denote evaluation matrix. It means to establish the fuzzy evaluation matrix to all factors:

$$R = \begin{bmatrix} R_1 \\ R_2 \\ \dots \\ R_7 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{15} \\ r_{21} & r_{22} & \dots & r_{25} \\ \vdots & \vdots & \vdots & \vdots \\ r_{71} & r_{72} & \dots & r_{75} \end{bmatrix} \quad (1)$$

More experts evaluate to each factor respectively to obtain veracious value. For example, 50 percent experts believe that the effective rows of code are exceeding 2 thousand rows. The effective row of code is between 1.5 thousand rows and 2 thousand rows, which are considered by 20% experts. Ten percent experts consider the effective rows of code are between 1 thousand and 1.5 thousand rows. The effective rows of code are between 0.5 thousand and 1 thousand rows, which are considered by 15% experts. The rows of code are less than 0.5 thousand rows, which are considered by 5% experts. So the fuzzy evaluation matrix to factor  $u_1$  is  $R = (0.5, 0.2, 0.1, 0.15, 0.05)$ .

#### 5) Fuzzy synthetic evaluating

$$P = AR = (a_1, a_2, \dots, a_7) \begin{bmatrix} r_{11} & r_{12} & \dots & r_{15} \\ r_{21} & r_{22} & \dots & r_{25} \\ \vdots & \vdots & \vdots & \vdots \\ r_{71} & r_{72} & \dots & r_{75} \end{bmatrix} = (p_1, p_2, \dots, p_5) \quad (2)$$

#### 6) Calculating final evaluation

The evaluation result is  $p_k$  according to the principle of maximal subjection degree, if  $p_k = \max(p_1, p_2, \dots, p_7)$ . The possibility is corresponding remark of  $p_k$ .

#### C. Software key rating based on the relative synthetic evaluation of bottom event

In order to guarantee the accuracy of defining for software rating, they are compared which obtained by relative synthetic evaluation of bottom event. The most reasonable one can be acquired.

Fix the system risk index using the serious grade definition of risk and possibility of risk occurrence which is obtained through the relative synthetic evaluation. At last, set the software key rating according to the system risk index and software controlling category. Software controlling category is relevant to the

software control program for system, and it is responsible for the real-time and compicacy of controlling. It can be ensured by designers on the basis of designing feature. When there is more than one rating for software, the top one is adopted.

### III. EXPERIMENT VALIDATIONS

By taking software of space payload system as example, the detailed process of the new method is illustrated. And the new means is compared with the method based on the vague set.

#### A. Fault tree foundation

The fault tree is established on the base of payload characteristic. Fault of control software for payload is regarded as top event. There are seven bottom events: they are  $X_1$ ,  $X_2$ ,  $\dots$   $X_7$  respectively. As shown in Figure 1.

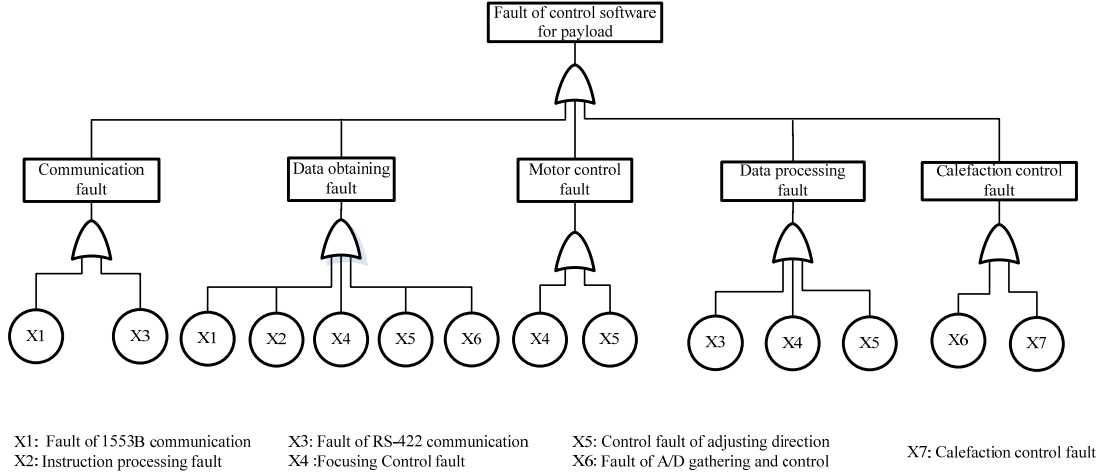


Figure 1. Fault tree for control software of space camera

#### B. Ascertaining software key rating

Table 1 is the values of seven evaluation factors and them Comment definition

The weight of being closely calculated is  $A = \{a_1, a_2, \dots, a_n\} = \{0.13, 0.14, 0.17, 0.11, 0.10, 0.17, 0.18\}$ . To take calculating 'fault of 1553B communication' as an example, the evaluation matrix is get through evaluating to each factor by ten experts.

TABLE I. DEFINITION OF FACTOR DEREFERENCING

Factor \ Y	more possible	very possible	possible	unlikely	impossible
$u_1$	$2 < u_1$	$1.5 < u_1 \leq 2$	$1 < u_1 \leq 1.5$	$0.5 < u_1 \leq 1$	$u_1 \leq 0.5$
$u_2$	$10 < u_2$	$8 < u_2 \leq 10$	$5 < u_2 \leq 8$	$1 < u_2 \leq 5$	$u_2 \leq 1$
$u_3$	$0.9 < u_3 \leq 1$	$0.8 < u_3 \leq 0.9$	$0.6 < u_3 \leq 0.8$	$0.4 < u_3 \leq 0.6$	$0 < u_3 \leq 0.4$
$u_4$	$0.9 < u_4 \leq 1$	$0.8 < u_4 \leq 0.9$	$0.6 < u_4 \leq 0.8$	$0.4 < u_4 \leq 0.6$	$0 < u_4 \leq 0.4$
$u_5$	$4 \leq u_5$	$u_5 = 3$	$u_5 = 2$	$u_5 = 1$	$u_5 = 0$
$u_6$	$6 \leq u_6$	$4 < u_6 \leq 6$	$2 < u_6 \leq 4$	$1 < u_6 \leq 2$	$u_6 \leq 1$
$u_7$	$6 \leq u_7$	$4 < u_7 \leq 6$	$2 < u_7 \leq 4$	$1 < u_7 \leq 2$	$u_7 \leq 1$

$$R = \begin{bmatrix} R_1 \\ R_2 \\ \dots \\ R_7 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{15} \\ r_{21} & r_{22} & \dots & r_{25} \\ \vdots & \vdots & \vdots & \vdots \\ r_{71} & r_{72} & \dots & r_{75} \end{bmatrix} = \begin{bmatrix} 0 & 0.1 & 0.1 & 0.4 & 0.4 \\ 0 & 0 & 0.1 & 0.7 & 0.2 \\ 0 & 0.1 & 0.2 & 0.6 & 0.1 \\ 0.8 & 0.1 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0.6 & 0.2 & 0.1 & 0.1 \\ 0 & 0.7 & 0.15 & 0.1 & 0.05 \end{bmatrix}$$

$$P = AR = (p_1, p_2, \dots, p_5) = (0.11, 0.227, 0.133, 0.387, 0.123)$$

The evaluation result is  $P_{x1} = p_4 = 0.387$  according to the principle of maximal subsection degree. The corresponding comment is unlikely, so the possibility of happening is unlikely. On a similar plan, other results can be calculated respectively.  $P_{x2} = p_3 = 0.79$ ,  $P_{x3} = p_2 = 0.44$ ,  $P_{x4} = p_3 = 0.25$ ,  $P_{x5} = p_3 = 0.24$ ,  $P_{x6} = p_4 = 0.13$ ,  $P_{x7} = p_5 = 0.20$ . The comments

are possible, very possible, possible, possible, unlikely and impossible.

TABLE II. DEFINITION OF RISK SEVERITY

Grade of risk severity	Risk definition
catastrophic	Payload breakdown during missions
severe	Payload unable work and unable obtain any valid remote image during missions
average	Have affection for accomplish remote sense tasks and remote image quality descending during missions
	Have affection for single task during missions

TABLE III. DEFINITION OF SYSTEM RISK INDEX

Happening possibility Serious grade of risk	more possible	very possible	possible	unlikely	impossible
catastrophic	1	1	2	3	4
severe	1	2	3	4	5
average	2	3	4	5	6
neglectful	3	4	5	6	7

The controlling category is II that is to say ‘some risk controlling action may be time crucial, have demand of critical time, but cannot exceed needful time for operator or the automatic respond time of system’ according to Table 4. The software key rating is C using the definition of Table 5.

TABLE IV. DEFINITION OF SOFTWARE CONTROLLING CATEGORY

Software controlling category	Definition
I	1 The risk is partial or all autonomously controlled by software
	2 There are multi-subsystem and interacting parallel processing or multi-interface
	3 Partial or all key function is time crucial
II	1 Risk is controlled and other safety system can be partial remission.
	2 It is a medium complex system which has no parallel processing and a small amount subsystem and/or some interfaces.
	3 Some risk controlling action may be time crucial, have demand of critical time, but cannot exceed needful time for operator or the automatic respond time of system
III	1 Provide redundant information resources of safety
	2 A little complex system that contains limited number interfaces
	3 relief system can respond in any critical time
IV	1 No controlling for risk
	2 It is a simple system which has 2-3 subsystem and a small amount interfaces.
	3 No demand for critical time

TABLE V. MATRIX DEFINITION OF SOFTWARE KEY RATING

Risk index Software controlling category	2	3	4	5
I	A	B	C	D
II	B	C**	D	D
III	C**	D	D	D
IV	D	D	D	D

	Partial functions fail to materialize and the accomplish task is effected during missions
neglectful	Have obstacle for accomplish task but have no influence on accomplish task and remote image quality affected slightly during missions

Similar to take calculating ‘fault of 1553B communication’ as an example, the serious grade of bottom event is catastrophic. The possibility is unlikely. The risk target is four according to the definition of system risk index (see Table 3).

The rating according to the other bottom event and top event is obtained as same (see Table 6).

TABLE VI. SOFTWARE RATING MATRIX TABLE OF PAYLOAD

Risk index Software controlling category	2	3	4	5
I				
II		C**	D	
III				D
IV				

So the rating is C\*\* which is taller than C and lower than B.

In fact, the design of software is finish. The quantification data can be collected. The experience we have gained over the engineer shows that the C\*\* is more appropriate than original rating. By comparison the new method and old one, the new method is obvious have the whip hand in efficiency and accuracy.

## SUMMARY

The method of key rating ascertainment for space software based on fuzzy synthetic evaluation and quantitative fault tree analysis is detailed expounded. The results indicate that the new method can ascertain the software key rating more accurate and reasonable. It offers the quantitative data for software management and test. Thereby guarantee the software quality finally.

## REFERENCES

- [1] Li Yang, Song Kefei, Application of Vague set to software rating ascertainment for space camera[J], The 2nd International Conference on Information Science and Engineering, 2010, 11.
- [2] DU Li, LIU Yu, HUANG Hong-zhong, PENG Wei, Aeroengine reliability allocation based on fuzzy prorating method and fuzzy synthetic assessment[J], Journal of Aerospace Power, 2009, 2 (in Chinese)

- [3] DU Li, ZHANG Han-liang, HUANG Hong-zhong, and LIU Yu , Combining Fuzzy Similar Product and Fuzzy Comprehensive Evaluation Method for Reliability Prediction of Diesel Engine[J] , Journal of University of Electronic Science and Technology of China,2012,7.
- [4] YUN Shaohui. Fuzzy synthetic evaluation for organizational learning ability based on AHP[J]. Journal of University of Electronic Science and Technology of China (Social Sciences Edition), 2007, 9(5): 26-28.
- [5] ZHANG Honglin, ZHANG Chunyuan,LIU Dong, FU Jian, A Method of Disjoint Quantitative Analysis for Dynamic Fault Tree[J], Journal of Computer Research and Development.2012,5.
- [6] WANG Jiaxu, ZHOU Qinghua, XIAO Ke, QIN Yi, HUANG Yanyan, Dynamic fault tree model analysis of systems subjected to incomplete common-cause failure, Engineering and Electronic.2012,5.
- [7] HUANG Yongqin, JIN Lifeng,LIU Yao, Current situation and trend of reliability technology in high performance computers[J]. Journal of Computer Research and Development,2010,47(4):589~584(in Chinese).