The Engineering Practice on Command Information System and Application Software

Hua Wang, Xian-Yu Li, Gang Xiang, Jian-Qun Zheng, Xiao-Kun Li
Department of Collectivity
The Fourth Research Institute of SAEA
Beijing, China
919wh_cn@sina.com

Abstract—This paper sets forth the technical reviews, question eliminating and configuration management of the Command Information System, depicts the measures taken in the system reliability, maintainability and supportability design; and the system availability models are established of full functions, main functions and lowest functions, moreover the system availability is quantitatively analyzed. Thenceforth a primary prospect is presented.

Keywords-system; application software; availability

I. INTRODUCTION

The availability of Command Information System is greatly helpful to the exertion of system functions and performance. During the development of the Command Information System, the technical and management responsibility are entirely carried out, the quality assurance system is founded, and the technical design and software engineering management is strengthened, systemic plan and implement are also carried through with respect to the availability management, design, analysis, and experimentation. Thus system availability capability is effectively ensured [1, 3].

II. SYSTEM AVAILABILITY MANAGEMENT

During the development of the system, combining many key processes at the elementary phase, the formal phase and the finalization phase, such as requirements management, project scheme, project tracking and supervision, subcontract management, quality assurance, configuration management, peer reviews and intergroup coordination, the following availability and quality management measures are taken.

A. System Technical Reviews

The system technical reviews are strengthened. The reviews are carried through of the elementary system, the formal system and the finalization system, requirement analysis, general and particular design, validation testing. Review team goes through the phase work and products, and form the review notion. And the quality of software design is noted, the process and outcome are noted, the problems of system reviews are solved and certificated by the superior before the next phase. Thus the cumulated effect of system bugs was well avoided.

B. Problems Disposal

The problems of system reviews, tests, and experimentations are solved. In the process of unit test, component test, validation test and system joint test, the closed loop policy of test implement-fault analysis-program modification-validity validation is strictly brought into effect. Thus the cumulated effect of system bugs was well avoided. Thus system availability capability is effectively improved.

C. Configuration Management

The software configuration management is strictly implemented. The function baseline, assignment baseline and product baseline are set up in the process of software development, and marker method is fixed on of the program, document, and their version. The development library, controlled library and product library is founded at project level and collectivity level, and the regulation of chick-in and check-out is established. The variations that surpass the baselines are actualized according to the variation control demand, and the configuration state is renewed in time. And software configuration is audited before software tests and delivery. Thus the software and system availability capability is strongly ensured.

III. SYSTEM AVAILABILITY DESIGN

General equipments and software are adopted in the platform of the Command Information System hardware and software, and mature techniques design, modularization design and redundancy design are introduced. The design of hardware and software and their interfaces meet the need of standardization, compatibility and expansibility. System has the capability of the fault identifying, examining, and isolating, and technical datum is of integrity and preciseness for the system use and maintenance. The software reliability, maintainability and supportability design is fully attached importance to in the process of software development, which provides a solid foundation for the system availability [1, 2, 4, 5, 6].

A. Software Reliability Design

Software reliability is ensured by the measures of software structure design, modularization design and program realization design.
Software structure of Command Information System comprises system application layer, special supportive layer, common supportive layer and basic data layer. System application layer aims at the concrete application, special supportive layer provides the service of basic information transacting, common supportive layer provides uniform sustentation for the service of information transacting, and basic data layer provides the service of data storage management.

Software modularization design is applied to the software development. Each module completes a special function of relative independence, and it is convenient for the module modification, testing and maintenance; the third class database is formed into logic whole, so that system data can be asynchronously renewed in real time to remain veracity and coherence; information shielding and localizing is adopted to cut down the falsity spread.

In the program realization, the references to the system globe variables are lessened, the process calls adopt function form, and validity of all the input parameters is examined; system abnormalities are monitored owing to outside or else modules, and disposal methods are reminded of the next step; standard interfaces of data call are employed to guarantee the stable mutuality of forward application program and backward database system.

B. Software Maintainability Design

General equipments and software are adopted in the platform of the system hardware and software, which is convenient for the system upgrade and maintenance.

System operation circs is noted by the means of log tool, and then system abnormality having uncertain causes can be ascended. Various kinds of information are fed back, so that operation personnel could be in control of matters and put up regulations at any moment. Database cases can be started by automation or manpower, and software system can be started over again when software failures result from the operation system. Software technical datum is complete, and the notation ratio of source program exceeds twenty percent.

C. Software Supportability Design

Software can be fast installed based on the uniform configuration management tool; related technical datum is offered such as software operation program and source code explanation, software user’s manual and product specification. Software operators are correspondingly trained to be able to use the software.

IV. SYSTEM AVAILABILITY ANALYSIS

A. Availability Model

Availability model indicates the variation relation of system availability to the reliability, maintainability, and supportability of components. Considering a repairable coherent system of n components, the states of component i and the system are 0-1 random variables \( X_i \) and \( X \), \( i = 1, \ldots, n \), \( X = (X_1, X_2, \ldots, X_n) \), and repair facility and maintenance personnel are numerous enough to go along repair matters when any component has malfunction. Supposing that the life distribution of component i is \( F_i(t) \) with the mean \( 1/\lambda_i \), and the distribution of fault-repair time is \( G_i(t) \) with the mean \( 1/\mu_i \); and these random variables are independent each other, malfunction components are repaired as new [2, 3].

Based upon renewal process theory, instantaneous availability and steady availability can be drawn, especially \( A_i = (1 + \lambda_i/\mu_i)^{-1}, i = 1, 2, \ldots, n \); and a coherent system can be depicted by structure function \( \varphi \) and reliability function \( h(p) = E\varphi(X) \). Thus the instantaneous availability and steady availability of the coherent system are respectively as below.

\[
A_i(t) = h(A_1(t), A_2(t), \ldots, A_n(t)) \quad (1)
\]

\[
A_s = h(A_1, A_2, \ldots, A_n) \quad (2)
\]

Common systems in use, such as series system, parallel system and vote system, are all regarded as coherent systems, and the formula of the availability can be actually drawn.

B. System Availability

On the basis of system architecture and use demand, the availability frame charts of the system full functions, main functions and lowest functions are respectively as Fig. 1, Fig. 2 and Fig. 3, where k/n denotes that subsystem performs when at least k of n terminals operate smoothly.

![Figure 1.](attachment:figure1.png)  the availability frame charts of system full functions

![Figure 2.](attachment:figure2.png)  the availability frame charts of system main functions

![Figure 3.](attachment:figure3.png)  the availability frame charts of system lowest functions

The availability mathematics model is as follows.

\[
A = T_U/(T_U + T_N) \quad (3)
\]

Thereinto A: availability; \( T_U \): up-time; \( T_N \): down-time.

The system availability of full functions, main functions and lowest functions are respectively as follows.
The ZH sub-system availability of full functions, main functions and lowest functions are respectively as follows.

\[ A_{ZH1} = \frac{T_{ZHU}}{T_{ZHU} + T_{ZHN}} \]  
\[ A_{ZH2} = \sum_{i=1}^{13} C_i A_{ZH1} (1 - A_{ZH1})^{3-i} \]  
\[ A_{ZH3} = \sum_{i=4}^{13} C_i A_{ZH2} (1 - A_{ZH1})^{3-i} \]

The QB sub-system availability of full functions, main functions and lowest functions are respectively as follows.

\[ A_{QB1} = \frac{T_{QBU}}{T_{QBU} + T_{QBN}} \]  
\[ A_{QB2} = A_{QB3} = \sum_{i=4}^{8} C_i A_{QB1} (1 - A_{QB1})^{4-i} \]

The TX sub-system availability of full functions, main functions and lowest functions are respectively as follows.

\[ A_{TX1} = \frac{T_{TXU}}{T_{TXU} + T_{TXN}} \]  
\[ A_{TX2} = A_{TX3} = \sum_{i=2}^{4} C_i A_{TX1} (1 - A_{TX1})^{4-i} \]

The HQ sub-system availability of full functions, main functions and lowest functions are respectively as follows.

\[ A_{HQ1} = \frac{T_{HQU}}{T_{HQU} + T_{HQN}} \]  
\[ A_{HQ2} = A_{HQ3} = \sum_{i=4}^{8} C_i A_{HQ1} (1 - A_{HQ1})^{4-i} \]

The ZB sub-system availability of full functions, main functions and lowest functions are respectively as follows.

\[ A_{ZB1} = \frac{T_{ZBU}}{T_{ZBU} + T_{ZBN}} \]  
\[ A_{ZB2} = A_{ZB3} = \sum_{i=4}^{8} C_i A_{ZB1} (1 - A_{ZB1})^{4-i} \]

V. AVAILABILITY TEST

During the development of the system, strict testing and experiments are carried through in the elementary phase and the formal phase with all the questions settled. In the system finalization test, tryout and third party testing, inspections mainly proceed in abnormality aspect besides routine aspect. That guarantees the continuous increasing of the system availability.

In the process of system joint test, terminals of the ZH sub-system, the QB sub-system and the TX sub-system have some ordinary faults individually that respectively last 20 minutes, 15 minutes, 5 minutes, 5 minutes, 10 minutes, with the continual task time 10 hours, 13 hours, 12 hours. Thus the system availability of full functions, main functions and lowest functions are respectively as follows.

\[ A_{S1} = A_{ZH1} \cdot A_{QB1} \cdot A_{TX1} \cdot A_{HQ1} \cdot A_{ZB1} \approx 0.998535 \times 0.999405 \times 0.999405 \times 1 \times 1 \]  
\[ = 0.997347 \]
\[ A_{S2} = A_{ZH2} \cdot A_{QB2} \cdot A_{TX2} \cdot A_{HQ2} \cdot A_{ZB2} \approx 1 \]
\[ A_{S3} = A_{ZH3} \cdot A_{QB3} \cdot A_{TX3} \cdot A_{HQ3} \cdot A_{ZB3} \approx 1 \]

By taking the measures in availability design, analysis, tests and availability management, the system operates steadily and reliably in practical use, which meets relative demand of system development.

VI. CONCLUSION

During the development of the Command Information System, the technical reviews, question eliminating and configuration management are entirely carried out, system availability design is implemented, and the system availability models are established of full functions, main functions and lowest functions, moreover the system availability is quantitatively analyzed.

As systems engineering technique and software engineering practice develop increasingly, the software process control could be optimized, the capability of system design and realization could be promoted, the development work and product could be improved on, and the system availability capability will increase extensively.
ACKNOWLEDGMENT

We are grateful to the experts and secretaries for their guidance, understanding and patience; they are a constant source of support and encouragement.

REFERENCES


