

A Test Process Model and Optimization Framework Based on Network Schedule

Litian Xiao^{1,2,*}, Mengyuan Li¹ and Nan Xiao³

¹Beijing Special Engineering Design and Research Institute, Beijing 10028, China

²School of Software, TNList, KLISS, Tsinghua University, Beijing 10084, China

³China Aerospace Academy of System Science and Engineering, Beijing 10048, China

*Corresponding author

Abstract—In order to shorten the test period of spaceflight articles at the launch site and improve facility utilization and launch rate, the paper proposes our research about test process model and optimization framework. The approach overlays test activity on activity-on-arrow for network schedule. The test process model is composed of the test network schedule. Based on the relativity of test items and input driving as well as output response, the test process optimization framework is formed. The test critical paths are dismantled and shortened as much as possible to shorten the entire test period. Through practice, our achievement effectively improves the test efficiency of flight articles at the launch site. It lays the foundation for constructing the integrated test system.

Keywords—aerospace test; network scheduling; test process model; critical path; optimization framework

I. INTRODUCTION

All mechatronics equipment such as cars, planes, train, launch vehicle, etc., need to be tested before they can be put into normal operation. Especially for the high risk and high-reliability systems of space articles, test verification is particularly critical. In general, the system units of space flight hardware are shipped separately and integrated at the launch site. Then full-system tests provide confidence that the system can accomplish the mission and launch technology states ^[1]. Test plan explicitly takes into account the dependencies of each test activity on other activities. Establishing a network schedule can determine test process definition and the test duration for test activities. A network schedule is a workflow diagram such as activity-on-arrow network diagram as shown in FIGURE I. At present, space launch missions are very intensive. As short as possible, test critical path is the key to shortening the entire test duration. It also is the key to enhancing the launch efficiency at the existing launch sites.

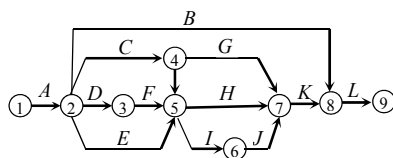


FIGURE I. ACTIVITY-ON-ARROW DIAGRAM FOR NETWORK SCHEDULES

II. RELATED WORK

A. On the Test Process Model

Test process model is a process model. Process description modeling is a hot research direction in process management system. According to their own needs and application, different scholars put forward different modeling methods ^{[2][3]}. The existing modeling methods are mainly based on interaction, activity, state transition, and relational capture. Its modeling methods do not take into account the impact of test schedule. It is not suitable for the actual physical test or simulation to improve the test schedule.

B. On Network Schedule

The activity-on-arrow diagrams for network schedule reflect the logical relation between the various activity processes of the project. It optimizes the durations, the costs and the resources of all activities by using the flexible time of each operation process. The activity-on-arrow diagrams for network schedule mainly adopt the ECRS' process optimization method (Eliminate, Combine, Rearrange, Simplify) ^{[4][5]}. The changes of test activity flow can involve the change of the technical states or configurations of the flight article and product or equipment. Mostly they are not acceptable to the flight projects.

C. On Aerospace Test

The research on aerospace test focuses on the planning resource construction and environment usage. The kinds of test resources continuously abundant at present, while the increased flexibility of a test system provides a greater number of choices of resources and routings ^{[6][7]}. The test activity schedule optimization of aerospace test system focuses the critical point to construct the resources and routings ^{[8]-[10]}. Planning resources is not involved in the schedule optimization of network schedule and related test items.

There are still some shortages of the related research in the field of aerospace testing. This paper presents an actual physical test perspective for test activity scheduling with shortening test critical paths. The research creates the test process model combined activity-on-arrow diagram with test activities and test items. We are working to break up the critical path of the test process, optimize the test process and shorten the test duration.

III. PRELIMINARY DEFINITIONS

A. Activity-on-Arrow Diagram for Test Networked Schedule

The tested system transforms the driving input into the response output according to the design state and configuration mapping. A test item needs to drive the tested system. Its result is the measurement data of the response output. A test system is composed of a tested system and test items.

Definition 1. According to the activity-on-arrow diagram, if test activity A , B and C are the serial test items labeled on the vector lines, LEN is defined as the time length, written as $LEN(\alpha_i)$, to complete a test item. $\alpha_i \in \{A, B, C\}$. The vector lines of unlabeled characters only represent the logical relationship, and the time length is zero. The time length to complete all the test activities is $T_L = LEN(A) + LEN(B) + LEN(C)$. If the test activities are parallel test items, $T_L = \text{MAX}(LEN(A), LEN(B), LEN(C))$. MAX is a maximum function. The test item of a test activity is the related subsystem testing.

B. Test Process Model Based on Activity-on-Arrow Diagram

On the basis of activity-on-arrow diagram for network schedule, each activity is defined as a test item on the diagram. Each start node of the arrow line is test input driving. Each terminal node of that is output responses or data.

Definition 2. Activity-on-arrow diagram is coded to form the set of M nodes, $N = \{n_1, n_2, \dots, n_M\}$, according to the sequences from left to right and from top to bottom (showed in Table 1.). Each node represents the input/output interfaces related the test items of a subsystem. It is the output results of the previous node and the input driving of the latter one. The start node of the whole diagram is the input driving of the test items, and the terminal node is the output results.

The typical logical relationship of test items is shown in Table 1. And other logical relationship can be dismantled into them. The start node or the end node can be the same one.

TABLE I. TYPICAL LOGICAL RELATIONSHIP BETWEEN TEST ITEMS

	Logical relationship between test items	The coded activity-on-arrow diagram	Equivalent Critical Path
1	To complete testing A , then testing B and C .		
2	To complete testing A and B , then testing C .		
3	To complete testing A and B , then testing D . To complete testing A , B and C , then testing E . To complete testing D and E , then testing F .		

Definition 3. The tested system features of a test item are decided by $\{PI, Tc, PO\}$, i.e. test items' action is controlled by input (PI), output (PO) and internal transition configuration (Tc) in a system. The description of test system can be defined as a tuple: $T_{sys} = \langle Tc, Tc_0, PI, PO, R_{PI} \rangle$, where Tc is the finite set of non-empty configuration function inside the tested system. With a test item, PI is the finite set of non-empty input, PO is the finite set of non-empty output, R_{PI} is the mapping transition set of an

input subset. $R_{PI} \subset \beta(PI) \ni \{r_i\}$, ($i=1,2,\dots$), which β is the transition function set of confirmed generalized verification. β filters an input subset that is effective in handling. r_i is the set of R_{PI} .

P^M describes the incomplete mapping of the tested system, i.e. the testing input is transited to testing output. $P^M(a, b): \beta(Tc) \times R_{PI} \rightarrow \beta(Tc) \times \beta(PO)$, where

$a \in \beta(Tc)$, $b \in \beta(Tc) \times \beta(PO)$. $Tc_0 \in \beta(Tc) \times \beta(PI)$, where Tc_0 is the initial state of the tested system.

C. The Relation of Test Systems

Definition 4. Test activities are connected by nodes. The relation R_{jl} of two test items can be represented as $(n_i \xrightarrow{\alpha_j} n_{i+1}) R_{jl} (n_k \xrightarrow{\alpha_l} n_{k+1})$, ($i, k \in (1, 2, \dots, N-1)$; $j, l \in (1, 2, \dots, L)$), where R_{jl} is combinational operator $\{\rightarrow \mid \cup \mid \cap \mid \oplus \mid \neg \mid \emptyset\}$.

A node is preceded by $?$ as an input and by $!$ as an output. Then $?n_i(\alpha_j)$ denotes an input of α_j at a node n_i . A test item is demonstrably described by $(?n_i \xrightarrow{\alpha_j} !n_{i+1})$ or a tuple (n, α, n') . It can be tagged with $P_j^M(?n_i, !n_{i+1})$ and briefly with P_j^M .

Definition 5. Let $P^M = [P_1^M, P_2^M, \dots, P_L^M]$, R_c is $L \times L$'s relation matrix $\begin{bmatrix} \emptyset & R_{21} & \dots & R_{L1} \\ R_{12} & \emptyset & \dots & R_{L2} \\ \dots & \dots & \dots & \dots \\ R_{1L} & R_{2L} & \dots & \emptyset \end{bmatrix}$. The test relation set can be briefly expressed as $P^M \circ R_c \circ (P^M)^T$. " \circ " is the relational composition operator.

IV. RELATIVITY IN TEST PROCESS MODEL

According to the definition of the relation R_{jl} associated nodes, the relativity of test item is divided into following four types.

(1) Independence, labeled R_{jl}^0 . There are no relation between two test items which can be performed separately. The input of a test item is only self-correlation.

$$P_j^M(?v_i, !v_{i+1}) R_{jl}^0 P_l^M(?v_k, !v_{k+1}) \xrightarrow{?n_k \not\rightarrow !n_{k+1}} (P_j^M(?v_i, !v_{i+1}), P_l^M(?v_k, !v_{k+1}))$$

(2) Uncorrelated, labeled R_{jl}^- .

The latter test item is unaffected by the result of the previous one. Two test items can be performed independently. The input of a test item could be regarded as self-correlation. $P_j^M(?n_i, !n_{i+1}) R_{jl}^- P_l^M(?n_k, !n_{k+1}) \xrightarrow{?n_k \not\rightarrow !n_{k+1}} (P_j^M(?n_i, !n_{i+1}), P_l^M(?n_k, !n_{k+1}))$

(3) Weak correlation, labeled R_{jl}^{\sim} .

The output result of the previous test item can exert influence on the result of the later one. However, as long as the output result of the previous test item is at normal or positive deviations (better than normal), it cannot influence on the result

of the later one. If it is a negative deviation, the result of the latter test item must be incorrect or uncertain.

$$\frac{P_j^M(?v_i, !v_{i+1}) R_{jl}^M P_l^M(?v_k, !v_{k+1})}{(?n_k \geq !n_{i+1}) \cup (?n_k \leq !n_{i+1})} \rightarrow P_j^M(?v_i, !v_{i+1}) \cup P_l^M(?v_k, !v_{k+1})$$

(4) Strong correlation, labeled R_{jl}^+ .

The output result of the previous test item determines the result of the later one. The next test item requires the actual values of the previous result as the input. Strong correlated path is consisted of continuous strong correlated test items.

$$P_j^M(?v_i, !v_{i+1}) R_{jl}^+ P_l^M(?v_k, !v_{k+1}) \xrightarrow{?n_k \geq !n_{i+1}} P_j^M(?v_i, !v_{i+1}) \times P_l^M(?v_k, !v_{k+1})$$

Therefore, according to the relativity, the critical paths can be dismantled into some strong correlated paths and separated test items. The shortest test period is determined by a maximum of the longest time length of the strong correlated critical path and max time length in the single tests.

Finally, the test process model can be abbreviated as

$$\cup P^M \circ R_c \circ (P^M)^T.$$

V. THE OPTIMIZATION FRAMEWORK FOR TEST PROCESS MODEL

A. The Transition of Strong Correlated Test Items

There are two cases of the relationship between two strong correlated test items. One case is that the response results are output at one-time after finishing the previous test item. The latter test item simultaneously receives the output as its input. We call this case synchronous testing.

Another case is that the previous test item output the grouping responses at different time intervals. The previous test item has not completed, the latter one start performance and testing after it receive the first group or several groups of the previous outputs. The subsequence groups of the previous outputs continue to be as the inputs of the latter test item until finishing the previous. We call this case asynchronous testing. FIGURE II shows the output of time progresses along from n_1 to n_2 which represents the testing process.

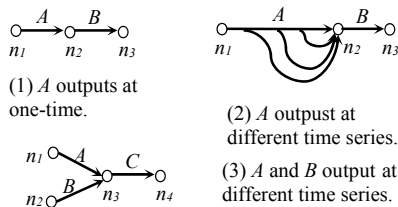


FIGURE II. THE INPUT/OUTPUT OF SYNCHRONOUS AND ASYNCHRONOUS TESTING

B. Synchronous Testing for Strong Correlated Test Items

Under synchronous testing of test items, an execution of the test items on strong correlated path is a finite sequence $\langle ?n_1, r_1 \rangle$,

$\langle ?n_2, r_2 \rangle, \dots, \langle ?n_i, r_i \rangle, \dots$, where $\langle ?n_i, r_i \rangle \in (P_i \times \cup.Tc)$ for each i and j . $\forall i \& j, \exists$ a transition $(?n_i, \varphi, !n_{i+1}) \in R_{PI} : (r_i, r_{i+1}) \models \varphi$. A path is a finite sequence $\rho = (?n_1, \varphi_1, !n_2), (?n_2, \varphi_2, !n_3), \dots, (?n_i, \varphi_i, !n_{i+1})$...of transitions from the initial node to the end of strong correlated path. A compound transition $(?n_1, \varphi, !n_k)$ can be defined by $(\sigma \circ \tau)(Tc, Tc') = \exists Tc''. \sigma(Tc, Tc'') \cap \tau(Tc'', Tc')$, where $\varphi = \varphi_1 \circ \varphi_2 \dots \circ \varphi_j$ and “ \circ ” is the relational composition operator. σ 's definition is consistent with τ .

Therefore, the inputs of the start node on the strong correlated path are synchronously transformed to the inputs of each test item by mapping transition. Each test item can be synchronously performed by the transformed inputs. The strong correlated path can be cut into the single test items which are performed with the transformed inputs. In the case of sufficient test resources, the maximum of test time lengths in the single test items determines the shortest test period.

C. Asynchronous Testing for Strong Correlated Test Items

Under asynchronous testing of test items, test items may have the delayed inputs/outputs that are asynchronous input/output. The next test item is performed according to the previous outputs by triggering on a first-come and first-served basis. If the previous output sequences are discrete on time, the strong correlated test items can be disposed of according to the below definition and rules.

Definition 6. Under the condition of asynchronous input and output, $?n = !n(1)!n(2) \dots !n(t)$ is the relation between input and output at different time t on n . Given path ρ of the asynchronous testing path, $\Delta(\rho)$ is the smallest set of the path defined as the below rules.

There is $\rho \in \Delta(\rho)$. If $\rho_1 n^1 n^2 \rho_2 \in \Delta(\rho)$, $n^1 \in P_O$ and $n^2 \in P_I$, then $\rho_1 n^2 n^1 \rho_2 \in \Delta(\rho)$. Given set π of asynchronous testing path, let $\Delta(\pi) = \cup_{\rho \in \pi} \Delta(\rho)$.

Given node set N_d belonged to ρ , $\Delta_{Nd}(\rho)$ is the smallest set of the asynchronous testing path. Definition 6 can be extended by the following:

1) $\rho \in \Delta_{Nd}(\rho)$. 2) If $\rho_1 n^1 n^2 \rho_2 \in \Delta_{Nd}(\rho)$, $\rho_1 n^2 n^1 \rho_2 \in \Delta_{Nd}(\rho)$ when:

- (a) n^1 and n^2 are at the same node, also $n^1 \in P_O$ and $n^2 \in P_I$.
- (b) n^1 and n^2 are at different nodes and either $n^1 \in P_O$ or $n^2 \in P_I$.

Given set π of asynchronous testing path, let $\Delta_{Nd}(\pi) = \cup_{\rho \in \pi} \Delta_{Nd}(\rho)$.

This extended definition can adapt to the general situation that n^1 and n^2 are at different nodes. Then n^1 and n^2 can be swapped as shown in FIGURE III.

FIGURE III (1), n^1 and n^2 are inputted. Since there is C channel between node n_1 and n_3 , the input orders of $?n_3$ in which they were sent by the transitions of B 's $?n_2$ and C 's $?n_1$ may be different.

FIGURE III (2), n^1 and n^2 are outputted. Since there is C channel between node n_1 and n_3 , the output orders of $!n_3$ in which

they were sent by the transitions of B 's $!n_2$ and C 's $!n_1$ may be different.

FIGURE III (3), n^1 is an output and n^2 is an input. If the output of C 's $!n_1$ is produced before the output of B 's $!n_2$, it is possible that the input orders of $?n_3$ firstly receives C 's output after it sends the input.

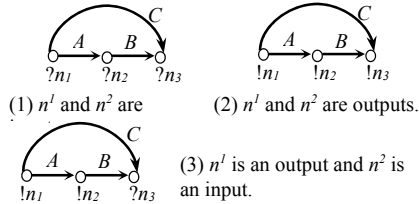


FIGURE III. n^1 AND n^2 HAVE THREE INTERCHANGEABLE TRANSFORMATION

If there is asynchronous testing path, a finite path $\Delta_{Nd}(\rho)$ has k items and an execution $\{<n_i, r_j>\}$, a compound transition $(?n_1, \varphi, !n_k)$ can be defined by $\Delta_{Nd}(\sigma \circ \tau)(Tc, Tc') = \exists Tc''. \Delta_{Nd}(\sigma(Tc, Tc'')) \cap \Delta_{Nd}(\tau(Tc'', Tc'))$, where $\varphi = \varphi_1 \circ \varphi_2 \dots \circ \varphi_j$ and " \circ " is the relational composition operator.

So the strong correlated test items can be cut into the separated parallel test items. Thus the whole test period will be shortened. Synchronous testing can be considered as a special case of asynchronous testing. The optimized test process model can be described as $\cup \Delta_{Nd}(P^M) \circ R_c \circ \Delta_{Nd}((P^M)^T)$, i.e. $\cup \Delta_{Nd}(P^M \circ R_c \circ (P^M)^T)$. Shortening the length of the strong correlated path is the critical to further shorten the test activity duration.

VI. CONCLUSION

If the relationship between test items is independence or uncorrelated, based on the test relativity in the test process model, separately to test each test item is clearly feasible. In the case of weak correlation between the test items, each test item can be independently tested. Optimizing the test process model makes the strong correlated test items in test system dismantled into shorter strong correlated paths and separated test items. It's feasible to be proved on theory, which limited space not to be carried out here.

Although the research imperfections need to be studied continuously, the model and the optimization framework have achieved good results. We have effectively reduced the full-system test period for flight articles at launch sites. The period is 40-60 days or longer in past. Now it can be shortened by 20-30 days. The achievement improves the utilization rate and the launch rate at launch sites. There are some imperfections, we are also committed to further research for improving the approach. From the preliminary results and the practice, the technical route for improving imperfections is feasible.

Through practice, this model not only contributes to shortening the test period but also lays the foundation for constructing the integrated test system. It can provide the basis on the quantitative model for the configuration of test resources.

ACKNOWLEDGEMENT

This work was funded by the project of General Technology on Test and Launch partially sponsored by the 973 program. We would like to thank Tsinghua's Prof. Yu Liu who provided us with the helpful suggestions and some research resources in the research.

REFERENCES

- [1] H Dezfuli, et al., NASA Systems Engineering Handbook, NASA USA, June 2007
- [2] MA Ramos, et al., Extending statecharts to model system interactions, J. Journal of Software Engineering Research & Development, 3 (1) (2015) 1-25.
- [3] A Natarajan, User-oriented modeling of scientific workflows for high-frequency event data analysis, C. IEEE International Conference on Data Engineering Workshops, 8-12(2013) 306-309.
- [4] Tan et al. A business process intelligence system for enterprise process performance management, J. IEEE Transactions on systems, man and cybernetics - Part C: Applications and Reviews, Vol. 38, No. 6 (2008).
- [5] M Fathi, M Ghobakhloo, A technical comment on "a review on assembly sequence planning and assembly line balancing optimisation using soft computing approaches", J. International Journal of Advanced Manufacturing Technology, 71 (9-12) (2014) 2033-2042.
- [6] George Tsakalidis, Kostas Vergidis, Towards a Comprehensive Business Process Optimization Framework, C. 2017 IEEE 19th Conference on Business Informatics (CBI), Vol.01(2017) 129-134.
- [7] Janis Grabis, Charu Chandra. Joint optimization of process design and operational policies. IEEE Engineering Management Review, Volume: 44, Issue: 3(2016) 32-45.
- [8] J Tong, Y Cai, A Method of Aerospace Test Task Scheduling Based on Genetic Algorithm, C. 2012 Spring Congress on Engineering & Technology, (2012) 1-4
- [9] Derek J. Seibel, Eric A. Mehiel, Task Scheduling through a Hybrid Genetic Algorithm, C. AIAA Modeling and Simulation Technologies Conference, Chicago. (2009) 1-13
- [10] Anneliese Andrews, et al., Testing Proper Mitigation in Safety-Critical Systems: An Aerospace Launch Application, C. 2015 IEEE Aerospace Conference, (2015) 1-19