
A Method to Solve Logical Loops in the GO-FLOW Methodology

Takeshi MATSUOKA

*Systems Engineering Division, Ship Research Institute, Ministry of Transport
6-38-1, Shinkawa, Mitaka, Tokyo 181-0004, Japan
E-mail:mats@srivot.go.jp, Fax:+81-422-41-3126*

Abstract

A method to solve logical loops has been developed in the framework of the GO-FLOW methodology. As a simple example system, we consider an engineering system with an operator or an automatic detector. If an operator or an automatic detector finds out the system does not work, he (or start mechanism) tries to start the system. The corresponding GO-FLOW chart contains a loop structure. The procedure to solve logical loops in the GO-FLOW, and the analysis result of example system are explained. The GO-FLOW becomes a flexible tool in analyzing any engineering system with the aid of this loop solving function.

1. Introduction

The authors have developed a system reliability analysis methodology GO-FLOW[1], which is a success oriented system reliability analysis method. A modeling technique produces a GO-FLOW chart, which represents system configuration, engineering function and system failure phenomena. Originally the GO-FLOW can not treat logical loops, so it was required to construct a GO-FLOW chart not to contain any loop structure. Sometimes we had to encounter the loop configurations in the process of making GO-FLOW charts. Some of these loop structures could be avoided by changing the interpretation of logical combinations between components[2], but there were inherently logical loops in some systems. So, the authors have developed a method to solve logical loops in the GO-FLOW methodology.

In this paper, the framework to solve logical loops in the GO-FLOW methodology is explained for an example engineering system.

2. Example system

As an example system, we consider a simple system shown in figure 1. A engineering system is started at specific time with a certain success probability of start

action. An operator or an automatic detector observes the success operation in successive time. If the operator or the detector finds out the system does not work, he (or start mechanism) tries to start the system at that time. The engineering system is suffered from component failures, so the probability of system operation decreases with time if the start actions are not taken after the initial start. Even if the perfect observation and start action is supposed, the probability of system operation decreases with time, because of the component failures.

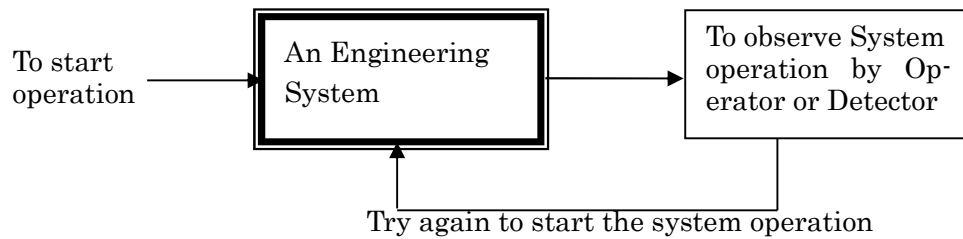


Figure 1. Example System.

3. GO-FLOW Chart

This system is modeled into a GO-FLOW chart as shown in figure 2. Six time points are defined in this example. Time point 1 is an initial state, and at time point 2 the operation of engineering system starts. Time point 3 is 10 hours after the beginning of the system operation (time point 2). Time points 4,5,6 are 20, 30, 40 hours after the beginning of the system operation, respectively.

In this figure, No.1 operator generates a signal which intensities are 1.0 at all the time points. No. 6 operator models the total component failures of the engineering system and the failure rates is given as $\lambda=3 \times 10^{-3}/h$. No. 3 operator generates a signal which gives time duration between successive time points. No.2 operator generates a signal which intensities are 0.0 at time point 1, and 1.0 at time points 2 to 6. Signals 2 and 6 are combined by an AND operator and generates the signal No.7 which represents the system operation from time point 2 to time point 6. No. 8 operator is a type 26 operator, that is, "Normally closed valve is opened by demand", and the open probability is given as 1.0 per demand. The success probability of system start at time point 2 is assumed as 0.6, and this is modeled by No.5 operator with the signal intensity 0.6 at time point 2.

The signal line 8 represents the success probability of system operation. Then, signal 9 represents the state that the system does not work. The probability that an operator or a detector finds out system condition properly is assumed as 0.9 and this is modeled by No.10 operator. The combination of operators Nos. 4 and 11 gives the condition that the observations are done after the time points 3. No. 12 operator

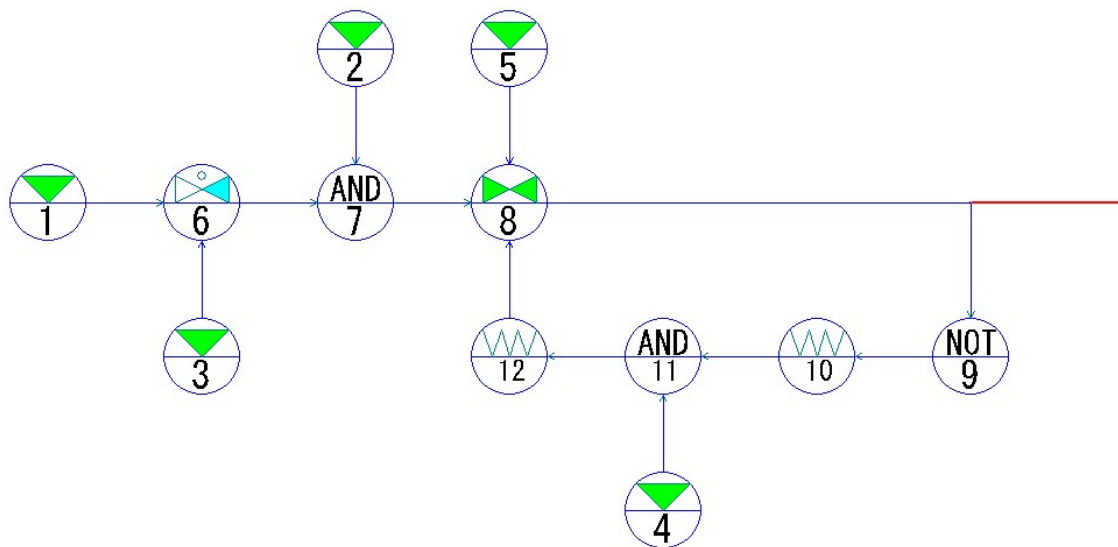


Figure 2. GO-FLOW chart of the system in figure 1.

models that the system starts with the probability 0.6 when start signal is given. The signal 12 is used as a sub-input signal to the No.8 operator and represents the start of system operation by additional start demands.

In this GO-FLOW chart, there is a loop structure.

4. Method to solve logical loops in the GO-FLOW

The authors have developed a framework to treat logical loops in the GO-FLOW methodology. In this framework, first construct a GO-FLOW chart without considering if there is any loop structure. After the construction of a chart, identify the portion of loop structures. Then, specify the cutting place(signal line) for each loop structure. The GO-FLOW analysis program automatically add a signal line at the cutting place of a loop, and iteration calculation is performed till the difference of signal intensities between cutting and additional signals becomes smaller than predefined value. The maximum number of iteration times can be settled by input data. We can select iteration time equals to one.

For the example system shown in figures 1 and 2, the signal line 12 is settled as a cutting place. A signal line (number 13) is automatically generated in the GO-FLOW analysis program and used for sub-input signal line to No.8 operator in this case, but the added signal line 13 is not written in figure 2. The difference between the intensities of signal lines 12 and 13 is detected and iteration calculation is performed till the difference becomes smaller than a predefined value (1.0×10^{-6} in the present analysis). Iteration was performed for 11 times in this example, and the required CPU time was 26seconds by a personal computer which has a Pentium II

400Mhz CPU and 3 GB memory.

5. Analysis result

The result of the analysis is shown in figure 3. The solid line shows the probability of system operation with additional start demands after the initial start. The broken line shows the probability of system operation only with initial start

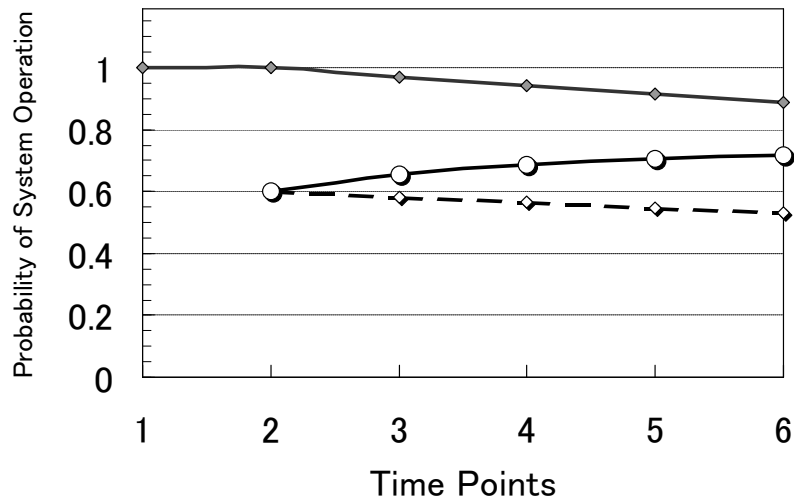


Figure 3. Analysis Result.

demand. The gray line shows the healthiness of engineering system except the start mechanism.

It is seen from this graph that the probability of system operation gradually increase with time because of additional start demands in case of detecting the system down condition.

6. Conclusion

A method to solve logical loops has been developed in the framework of the GO-FLOW methodology. As a simple example system, we considered an engineering system with an operator or automatic detector. The GO-FLOW chart of this example system contained a loop structure. The procedure to solve logical loops was explained in detail; identification of cutting place and iteration calculation. The example system could be analyzed with reasonable CPU time.

The GO-FLOW becomes a flexible tool in analyzing any engineering system with the aid of this loop solving function.

7. References

- [1]Matsuoka T. Kobayashi M., GO-FLOW:A New Reliability Analysis Methodology, Nuclear Science and Engineering, Vol.98, 64-78, 1988
- [2]Matsuoka T. Kobayashi M., An Analysis of a Dynamic System by the GO-FLOW methodology. Proceedings of the PSAM3, Crete, Greece, 1547-1552, 1996