Automated generation of a reliability model for a system undertaking phased missions.

S.J.Dunnett & K.S.Stockwell
Department of Aeronautical and Automotive Engineering, Loughborough University. Loughborough, Leics. LE11 3TU, U.K.

ABSTRACT: There are various mathematical models available to assess the reliability of a given system, these models relate the performance of the system to the performance of the components of which it is comprised and can be used to determine the failure probability or failure frequency of the system in question. Currently there is software available to perform the mathematical analysis of the model but its construction, which is used as input to the software, is undertaken manually. This is quite a lengthy process and can limit the usefulness of the model. One way of improving this situation would be to automate the construction process. In this work a procedure is developed to automatically generate a reliability model, based upon Petri Nets, for a system undertaking a phased mission.

1 INTRODUCTION

There are various techniques available to model the reliability of a system, such as fault trees, event trees, markov analysis. These models can be applied at the design stage to investigate alternative design options and influence the development of the system. They can also be used to prove that the system will perform to the required standard (perhaps satisfying the regulatory requirements where failures are safety related with the potential to result in fatalities) once its design has been finalised. However the most beneficial way to use the models is at the design phase when there is the most flexibility in changing the design in response to the predicted performance. As the development of the mathematical model is quite a lengthy process this can limit its usefulness. As during the time the models are being developed for a specific system design, and analysed, the system design independently progresses and evolves. Therefore the model and its influence lags behind the actual realised design. Another limiting factor is that design teams do often not have the expertise required to develop the models and it is therefore passed to a specialist group to perform the task. Hence resulting in a loss of overall control and project cohesion.

Over the years much work has been performed on the analysis of the models and this is now well developed and can be performed quickly. The area that still involves significant time and effort is the construction of the models. One way of improving this situation is to automate the construction process. This would make the analysis less complex enabling it to be performed by the design team. Automation also reduces the time of an analysis and can help prevent errors. Due to these benefits the subject has received some attention, most of which has concentrated on the Fault Tree approach. Various modelling approaches have been adopted to generate the tree. The most commonly adopted approaches include digraphs (Lapp & Powers 1977), decision tables (Salem et al 1977), transition tables (Taylor 1982) and mini fault trees (Kelly & Lees 1986). However no model has been developed which is appropriate for all conditions. Although Fault Tree Analysis cannot be used to accurately analysis many systems the automation of the other available techniques has received little attention. The aim of the work presented here is to develop a procedure to automatically generate a reliability model for a system undertaking a phased mission. Such a mission is made up of consecutive time periods, known as phases, within which the system may have to meet different requirements for successful completion of the phase. System failure during any phase will result in mission failure. The main techniques that have been used in solving phased mission problems are Fault Tree Analysis, Markov Analysis and simulation. Both fault trees and markov suffer the problem of developing very large models for such problems. Simulation techniques however are well suited to modelling such situations as their computational nature allows for complex scenarios to be considered. One method that allows for simple graphical

representation as well as significant modelling power is the Petri net.

2 PETRI NETS

A Petri net is a bipartite directed graph with two types of node: places, which are circular, and transitions, which are represented by bars. Places link only to transitions, and vice versa, using directed arcs. It is possible for a place to have several arcs to, or from, the same transition, which is condensed down into a single arc with a weight, or multiplicity, and denoted by a slash through the arc with a number next to it. If there is no lash the multiplicity is one. Tokens, or marks, reside within places and are passed between them by the switching of transitions. It is this switching of the transitions which represents the dynamic behaviour of the Petri net model. The net marking is a term given to the distribution of the tokens throughout the whole Petri net, and each form of it represents a different system state. It is this which is of interest to the analyst.

An example of transition switching is given in Figure 1.

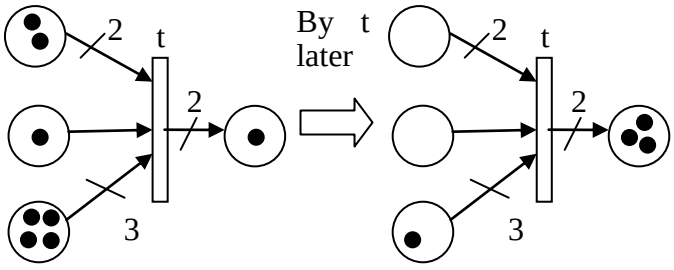

Figure 1. Transition enabling and switching

The figure shows a transition that has three places as inputs. A transition is said to be enabled when all the input places contain at least the weight number of tokens. In Figure 1 this is two for the top place, one for the middle place and three for the lowest place. The transition has a time delay t associated with it, this forces the transition to postpone switching for a period of time t upon being enabled. This delay can be deterministic, sampled from a distribution, or zero, in which case the transition is represented by a solid bar. Once the time period has passed and the transition remains enabled, the switching takes place. This process removes the number of tokens in each input place corresponding to the multiplicity of the relevant arc and creates the weight number of tokens in each output place. This is shown in Figure 1 where the switching removes two, one and three tokens from the input places and deposits two tokens in the output place.

3. AUTOMATIC GENERATION OF A PETRI NET MODEL OF SYSTEM RELIABILITY.

The aim of the procedure outlined in this work is to automatically generate a Petri net model to determine the reliability of a system performing a phased mission. The procedure is based on the following steps:

3.1 *Requirements.*

In order to generate a model, information about the system and its usage is required. This information falls into the categories of: component models, system topology, phase models, initial and starting conditions and failure conditions. The component models are in the form of decision tables which give a description of how the input and internal operational modes of each component influence the output states of that component. Operational mode tables, similar to the state transition tables used by Majdara and Wakabayashi 2009, are also adopted to model components with different operating modes. The system topology diagram describes how the components are linked together. The phase models describe the different phases that the system undergoes and includes the length of each phase and any conditions that must be satisfied in the phase and to make a transition between phases. The initial conditions are any conditions that the components must satisfy at the start of the mission. The system may also require certain conditions to be satisfied to start the system. The failure conditions are the failure modes of the components and the system. As the procedure is currently in the stages of being developed to automatically construct the model no component failure or repair data is yet necessary. This will be included at a later date.

3.2 *Model construction.*

In order to automatically generate the reliability model from the system description four distinct PN's have been identified, component nets (CPN), circuit nets (CiPN), system nets (SPN) and phase nets (PPN). The CPN's model the components failure and repair. The CiPN represent any circuits that are identified in the system and are used to determine whether current is being passed. Circuits are defined as a path starting and ending at the same component which passes current. These are obtained by starting at a power supply and tracing through the topology diagram. The SPN models the interactions between the components within the system and are developed using the decision tables, operational mode tables and the topology diagram. The PPN governs the phase progression and mission completion or abandonment. The different PN's interact by arcs linking

places and transitions and these arcs are the mechanism for passing information between the nets.

3.3 *Model simulation*

The PN's described above will be used to simulate the system reliability. Software is currently being developed to perform this task. The steps to be undertaken are:

1 From the initial conditions of the system place tokens in the relevant places within the SPN.
2 Place a token in the place representing phase 1 in the PPN.
3 Randomly sample failure and repair times for the components from the relevant distribution.
4 Search through each of the immediate transitions in the CPN, SPN and PPN and determine if any are enabled, if so fire them.
5 If the operating mode of a component changes then check the CiPN and place a token in C/NC in the out place of the component.
6 Repeat step 4.

When any transition is fired test if any of the following conditions are satisfied:

– a phase transition condition is satisfied. If mission has finished, failure or success, log results and start new simulation.
– In phase conditions are satisfied for current phase. Check for next timed transition and fire.

4 EXAMPLE

In order to demonstrate the procedure it is applied to a pressure tank control system.

4.1 *System description.*

The aim of the system, shown in Figure 2, is to control the filling and emptying of the tank. Initially the system is considered to be in a dormant state and thus de-energised. The switch (S1), the relay contacts (RC) and the timer contacts (TC) are open, the switch (S2) is closed and the tank is empty. Depressing S1 provides power to the timer relay (TIM) which results in the closure of its contacts, TC and the start of the timer mechanism. TIM self latches when S1 opens when released, and power is also supplied to relay (R) resulting in contacts, RC, closing which starts the motor (M) and hence pump (P). The tank (T) starts to fill. After a time t_1 the contacts TC open, relay R de-energises and its contacts, RC, open thus removing power from M and hence P. When TIM is de-energised the timer clock is reset. The operator (OP) will notice the tank pressure by the pressure gauge (PG) and will open the valve (V) to empty the tank. After a time t_2 the tank will have emptied sufficiently for filling to start again by OP pressing switch S1 and closing the valve. Switch S2 is a safety mechanism built into the system so that in the event of a failure occurring and the tank overfilling, the operator, who will be alerted by PG, can stop the pump by opening S2 hence denying power to R.

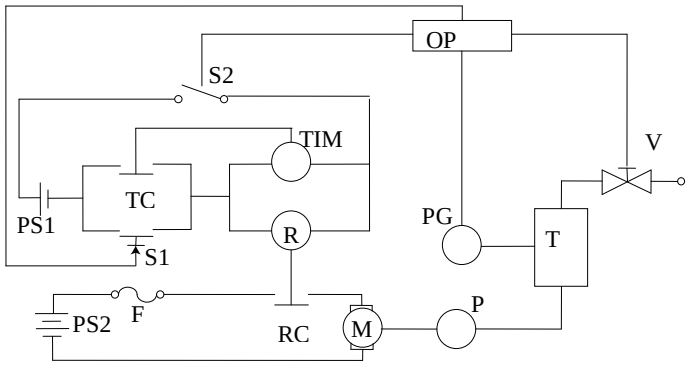

Figure 2. System schematic

The system described operates in three distinct phases:

1 Start up. This is a discrete phase only occurring momentarily when S1 is pressed.
2 Fill up, length t_1. In this phase, relay contacts TC and RC are closed, S1 is open, and V is closed. The tank will be filling.
3 Empty, length t_2. In this phase, relay contacts TC and RC are open, S1 is open, and V is open. The tank will be emptying.

A reliability model is automatically constructed by the following steps.

4.2 *Inputs.*

A topology diagram for the system is constructed as shown in Figure 3.

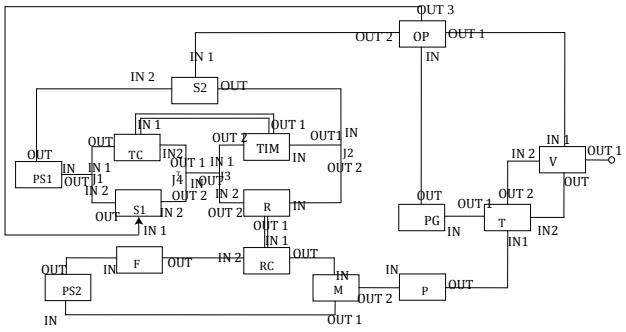

Figure 3. System topology.

The failure modes considered for the components in the system is shown in Table 1. Failure and repair rates have not been included as the example has been taken to demonstrate the construction process. Decision tables relevant to the components in the system are then constructed or obtained from a library of such tables that have been developed for the most common components.

Table 1. Component failure modes.

Component	Failure Mode	Description
Power Supply n PSn	PSn_F	No Power
Switch n, Sn	Sn_FO	Switch n failed open
	Sn_FC	Switch n failed closed
Timer relay	TC_FC	Contacts failed closed
TIM	TC_FO	Contacts failed open
	TIM_F	Relay failed de-energised
Relay,R	RC_FC	Contacts failed closed
	RC_FO	Contacts failed open
	R_F	Relay failed de-energised
Fuse, F	F_F	Fuse broken
Motor, M	M_F	Motor broken, pump not Started
Pump,P	P_F	Pump broken, no pumping
Valve, V	V_FC	Valve failed closed
	V_FO	Valve failed open
Pressure Gauge PG	PG_F	Fails to register pressure
Operator , OP	OP_F	Operator fails to take action
Tank,T	T_F	Significant leak in tank

As an example of how to construct such a table consider the pump (P), which has one failure mode in this example. From the topology diagram it can be seen that pump has one input from the motor and one output to the tank. The input has two possible states ON or OFF. The decision table, Table 2, considers all the possible combinations of the input and all possible states of P and the effects these will have on the output. In the table FL, NFL denote flow and no flow respectively. Considering the causes of flow in the output, from Table 2 it can be seen that only row 1 gives this result. This is the case when IN=ON and the pump is working. The combinations considered in rows 2 and 3 result in no flow in the output. The sign '-' in the input or state column indicates that the value of this particular variable does not affect the output in this case. For example in row 2 of table 2 if the pump is failed then the output to the tank will be no flow regardless of the input from the motor.

Table 2. Decision table for pump.

	In	State	Out
1	ON	W	FL
2	-	F	NFL
3	OFF	-	NFL

For some components the output will also be dependent upon time, this information is contained in the phase descriptions and is accounted for in the decision tables by adding an extra column for time.

For example, for TIM, the output to the contacts, OUT1, is dependent upon time. This is shown in table 3 where in row 1 for $t<t_1$ OUT1=EN, but for $t \geq t_1$, row 2, OUT1=DE for the same input and component state.

Table 3. Decision table for the timer relay.

	Time	IN	State	OUT1	OUT2
1	$<t_1$	C	W	EN	C
2	$t \geq t_1$	C	W	DE	C
3	-	-	F	DE	NC
4	-	NC	-	DE	NC

In the table EN and DE denote energised and de-energised respectively and C and NC denote current, no current.

Also some of the components in the system have different operational modes, for example, the switches have two different modes, open or closed. The change in mode will be a result of a change in input. For these components the change in mode has been modelled using operational mode tables as well as decision tables. For example the operational mode table for switch S2 (and Valve V) is shown in table 4. Both of these components have two modes, opened or closed, and their mode is changed by the operator through IN1, see Figure 3. As shown in table 1 the components have two failure modes, FO and FC. In the operational mode table all possible combinations of the current mode (mode 1), the input and the component state are considered and the effect on the mode (mode 2) determined. For example, if the current mode is closed, the input is open and the component is working (row 3) then the new mode will be open. In the table the notation NA is used for no action from the operator.

Table 4. Operational mode table for S2 and V.

	Mode 1	IN1	State	Mode 2
1	Closed	-	FC	Closed
2	Closed	CL	-	Closed
3	Closed	OP	W	Open
4	Closed	NA	-	Closed
5	Open	-	FO	Open
6	Open	OP	-	Open
7	Open	CL	W	Closed
8	Open	NA	-	Open

Mode 2 is then used as the mode in the decision table. For example the decision table for switch is given in Table 5.

Table 5. Decision table for S1, S2, TC, RC.

	In2	Mode	Out
1	-	Open	NC
2	NC	-	NC
3	C	Closed	C

The remaining operational mode tables and decision tables are shown in tables 6- 16.

Table 6. Operational mode table for S1

	Mode 1	IN1	State	Mode 2
1	Closed	-	FC	Closed
2	Closed	CL	-	Closed
3	Closed	NA	W	Open
4	Open	-	FO	Open
5	Open	NA	-	Open
6	Open	CL	W	Closed

Table 7. Operational mode table for TC, RC.

	Mode 1	IN1	State	Mode 2
1	Closed	-	FC	Closed
2	Closed	EN	-	Closed
3	Closed	DE	W	Open
4	Open	-	FO	Open
5	Open	DE	-	Open
6	Open	EN	W	Closed

Table 8. Decision table for PS_n, n=1,2

	In	State	Out
1	C	W	C
2	-	F	NC
3	NC	-	NC

Table 9. Decision table for R

	In	State	Out1	Out2
1	C	W	EN	C
2	-	F	DE	NC
3	NC	-	DE	NC

Table 10. Decision table for the junctions 1-4.

	In1	In2	Out1	Out2
1	C	-	C	
2	-	C	C	
3	NC	NC	NC	
4	C		C	C
5	NC		NC	NC

Table 11. Decision table for F.

	In	State	Out
1	C	W	C
2	-	F	NC
3	NC	-	NC

Table 12. Decision table for M.

	In	State	Out1	Out2
1	C	W	C	ON
2	-	F	NC	OFF
3	NC	-	NC	OFF

Table 13. Decision table for T

t	In1	In2	State	Out1	Out2
-	FL	OP	W	CONST	FL
-	FL	CL	W	INC	NFL
-	NFL	CL	W	CONST	NFL
$\leq t_1$	NFL	OP	W	CONST	NFL
$t_1<t\leq t_1+ t_2$	NFL	OP	W	DEC	FL
$\leq t_1$	-	-	F	CONST	NFL
$t_1<t\leq t_1+ t_2$	-	-	F	DEC	NFL

Table 14. Decision table for PG

	t	In	State	Out
1	$< t_1$	-	W	LPR
2	t_1	Const	W	LPR
3	t_1	Inc	W	HPR
4	-	Dec	W	LPR
5	$t_1 < t < t_1+ t_2$	Const	W	HPR
6	$t_1 < t < t_1+ t_2$	Inc	W	VHPR
7	-	-	F	NR

Table 15. Decision table for OP

t	In1	State	Out1	Out2	Out3
0	LPR	W	CL	NA	CL
$0<t<t_1+ t_2$	LPR	W	NA	NA	NA
-	HPR	W	OP	NA	NA
-	VHPR	W	NA	OP	NA
-	-	F	NA	NA	NA
0	NR	W	CL	NA	CL
$0<t<t_1+ t_2$	NR	W	NA	NA	NA

Table 16. Decision table for V

	In2	Mode	Out1	Out2
1	-	Closed	NFL	CL
2	NFL	Open	NFL	OP
3	NFL	Closed	NFL	CL
4	FL	Open	FL	OP

In the tables LPR, HPR and VHPR denote low pressure, high pressure and very high pressure respectively, NR denotes no reading and NA denotes no action.

The initial conditions are that switch 2 and Valve are closed and that switch 1, relay R contacts and timer relay TIM contacts are open, i.e.

S2_Mode=Closed,
V_Mode=Closed,
RC_Mode=Open,
TC_Mode=Open,
S1_Mode=Open

The system is started when the operator closes switch 1, i.e. S1_IN1=CL

The failure modes of the system have been broken down into ‘system overfill’ and ‘others’ as it is considered that overfill will be the most serious consequence. If overfill occurs the system can be successfully shutdown or continue. The method could be adapted to consider other failure modes. In the algorithm these failure states have been modelled as separate phases, The mission is considered to be a success if the phases 1-3 are completed. Hence there are eight phases to consider.

1 Start up
2 Fill
3 Empty
4 System overfill
5 System shutdown due to overfill
6 System overfill and no shutdown
7 Failure of the system other than overfill.

8 Mission success

From the system description the phase transition conditions can be determined as shown in table 17.

Table 17. Phase transition conditions

	t	From Phase	To Phase	Condition
1	0	1	2	TC_MODE= Closed
2	0	1	7	TC_MODE = Open
3	t_1	2	3	PG_OUT = HPR
4	-	2	4	PG_OUT = VHPR
5	t_1	2	7	PG_OUT = LPR
6	-	2	7	RC_MODE=Open
7	t_1+t_2	3	8	PG_OUT = LPR
8	-	3	4	PG_OUT = VHPR
9	t_1+t_2	3	7	PG_OUT = HPR
10	-	4	5	RC_MODE=Open
11	-	4	6	RC_MODE=Closed

The in phase conditions for the operational phases are:

1: S1_mode=closed,
2: T_IN1=FL, V_mode=closed,
3: T_IN1=NFL, V_mode=open.

4.3 *Petri Net development*

The different nets identified earlier, CPN, CiPN, SPN and PPN for this example are considered below.

4.3.1 *Component Petri Nets*

These are elementary nets for this simple no repair example and can be generated from the component descriptions. For the majority of the components they just consist of two places to define a working and failed state and a transition to model the time to failure, t_F, Figure 4. For some components the failed state will be dependent upon the current operation mode, the CPN's for such components in this example are shown in Figure 5.

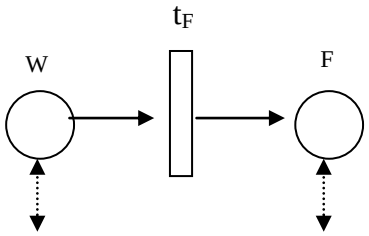

Figure 4. CPN's for PS1, PS2, TIM, R, F,M,P,PG,OP and T.

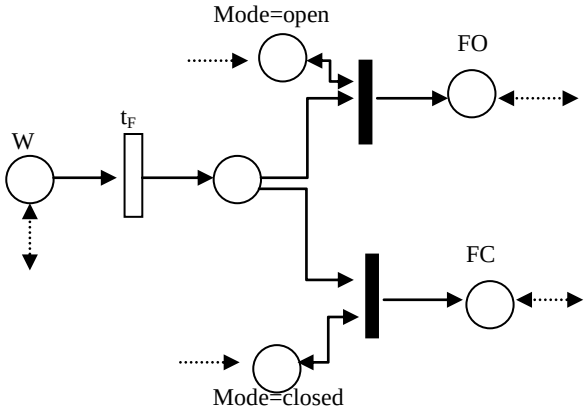

Figure 5. CPN's for S1, S2, V, TC and RC.

4.3.2 *Circuit Petri Nets*

In this example there are five circuits, C_1= {PS1, S2, J2, TIM, J3, J4, TC, J1, PS1}, C_2= {PS1, S2, J2, TIM, J3, J4, S1, J1, PS1}, C_3= {PS1, S2, J2, R, J3, J4, TC, J1, PS1}, C_4= {PS1, S2, J2, R, J3, J4, S1, J1, PS1}and C_5= {PS2, F, RC, M, PS2}. Circuit Petri nets are developed by considering the components in each circuit and using the decision tables to determine when they pass, or do not pass, current. As an example the Petri nets for current in circuit 1 and no current in circuit 1 are shown in figure 6.

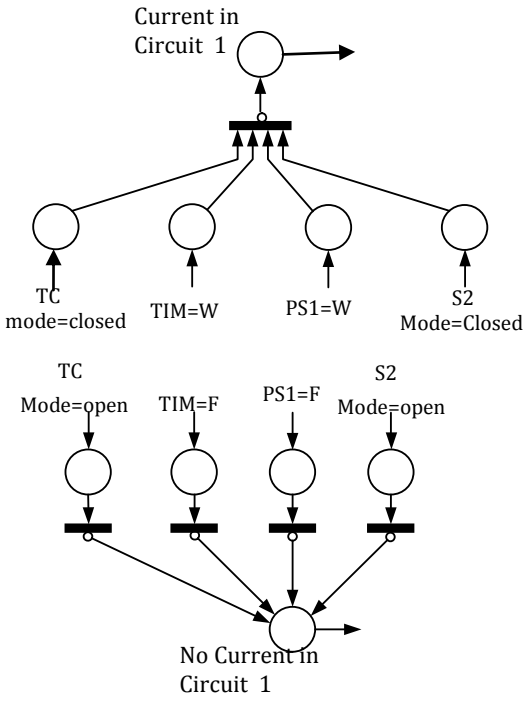

Figure 6. Petri nets for current in circuit 1 and no current in circuit 1.

4.3.3 *System Petri Nets*

From the information contained in the decision tables, Petri nets can be generated that model the effects upon the component output of the various combinations of inputs and component states. For example, figure 7 shows the Petri net generated from

the operational mode, and decision, table for S1, tables 6 and 5 respectively. Row 1 of table 6 states that if the mode is closed and the state is FC then mode remains closed, this is shown in the Petri net by the transition which has inputs 'FC' and 'mode=closed' and outputs to 'mode=closed'. The other rows are represented in a similar way. As switch S1 is a push button switch which opens when released the net also includes a transition, of delay δ, which has as input place 'IN1=CL' and output place 'IN1=NA'. The nets generated in this way are linked to form a SPN from the information in the topology diagram. Part of this net is shown in Figure 7. The dashed arcs in the figures represent links from other nets, CPN's, CiPN's or PPN's, or from elsewhere in the SPN.

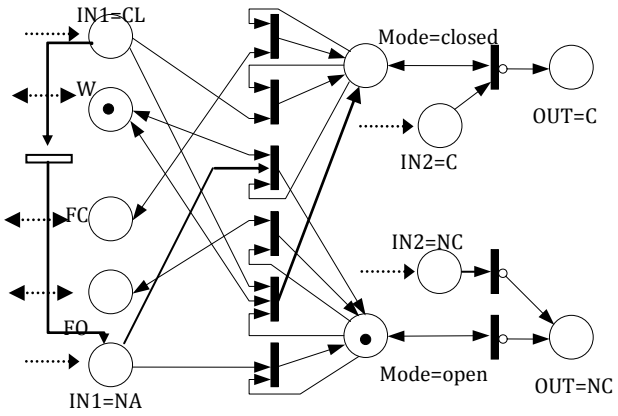

Figure 6. Petri net generated from tables 6 and 5.

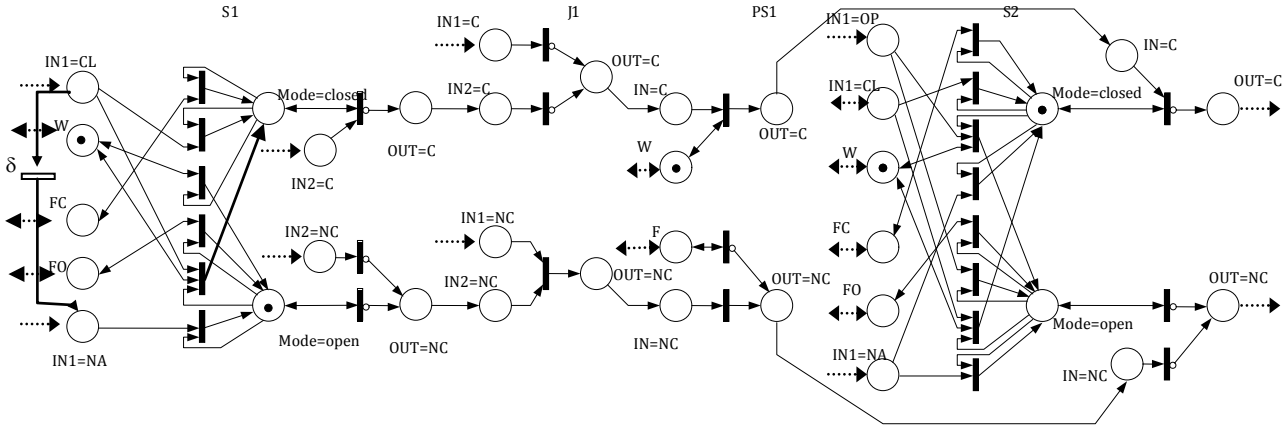

Figure 7. Part of the SPN

4.3.4 *Phase Petri Net*

The PPN controls the sequence of phases and failure, or success, of the mission and is generated from the phase information and the phase transition information.

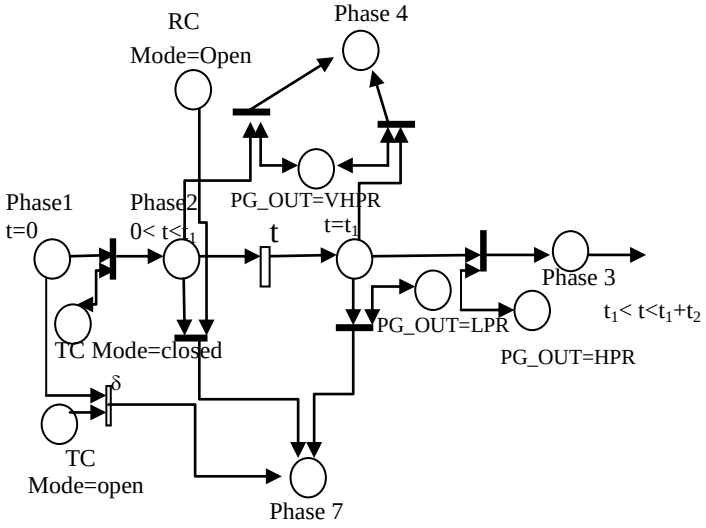

Figure 8. Part of the PPN for the example.

Part of the PPN for the example is shown in Figure 8. From the phase transition table, table 17, it is known that the transition between phase 1 and 2 occurs if TC mode=closed and the transition between 1 and 7 if TC mode=open. These transitions are seen in Figure 8 when tokens are in the places representing phase 1 and TC mode=closed and TC mode in open respectively. Paces in this net are also linked to places in the SPN.

5 SIMULATION MODEL.

Having generated the model as detailed above, the steps outlined in section 3.3 will be undertaken in order to predict the reliability of the system. This is the current area of the work being undertaken.

6 CONCLUSIONS

In tnis work a procedure has been outlined to automatically generate the reliability model for a system performing a phased mission. Employing the system topology diagram and decision and operating mode tables Petri nets are developed to model the system. Phase information is used to construct Petri nets that control the sequence of phases. For the systems considered to date Petri nets have also been developed to model the circuits within the systems. As this work is extended and a wider variety of systems are consiederd it it anticipated that the procedure will be adapted to deal with systems without circuits.

The model developed can then be used to estimate the system reliability and the procedure to achieve this has been outlined in this work This stage of the work is currently in process.

7 REFERENCES