This item was submitted to [Loughborough's Research Repository](#) by the author.
Items in Figshare are protected by copyright, with all rights reserved, unless otherwise indicated.

Modelling service support system reliability

PLEASE CITE THE PUBLISHED VERSION

<http://www.ifac-control.org/publications>

PUBLISHER

© IFAC

VERSION

AM (Accepted Manuscript)

LICENCE

CC BY-NC-ND 4.0

REPOSITORY RECORD

Reed, Sean, J.D. Andrews, Sarah J. Dunnett, Magnus Karlberg, Lennart Karlsson, and Magnus Lofstrand.
2012. "Modelling Service Support System Reliability". figshare. <https://hdl.handle.net/2134/11268>.

This item was submitted to Loughborough's Institutional Repository (<https://dspace.lboro.ac.uk/>) by the author and is made available under the following Creative Commons Licence conditions.

For the full text of this licence, please go to:
<http://creativecommons.org/licenses/by-nc-nd/2.5/>

Modelling Service Support System Reliability

Sean Reed*, John Andrews*, Sarah Dunnett**, Magnus Karlberg***, Lennart Karlsson***, Magnus Löfstrand***

**Nottingham Transportation Engineering Centre, University of Nottingham, Nottingham, U.K.*

***Department of Aeronautical and Automotive Engineering, Loughborough University, Loughborough, U.K.*

**** Division of Computer Aided Design, Luleå University of Technology, Luleå, Sweden*

Abstract: Functional Products, where the customer pays for the function and availability of a product instead of the product itself, are increasingly popular in capital intensive industries such as aerospace. Such products are integrated systems involving the combination of hardware and service support systems. The reliability prediction and optimisation of the service system that supports the hardware availability is essential to the feasibility of the product. These systems consist of maintenance procedures and resource provisions. Simulation based techniques are presented in this paper to analyse the reliability of support systems and their application is demonstrated through a simple example.

Keywords: Reliability analysis, Simulation, Petri-nets, Maintenance, Functional Products

1. INTRODUCTION

A Functional Product is an integrated system consisting of hardware and support systems. The customer of a Functional Product purchases the function that the product supplies, whilst the supplier retains ownership of the product and guarantees a certain level of availability. The supplier provides a service support system that consists of the services that ensure the guaranteed level of availability is achieved. This type of performance based contracting (PBC) is commonly known as “power by the hour” in the private sector and as performance-based logistics (PBL) in defence contracting. They offer benefits to both the supplier and customer and their development requires close collaboration between the two parties. A review of the literature on service design is given in Alonso-Rasgado et al (2004) and a generic methodology for the design of a functional product is presented.

The service support system is crucial in providing a stated level of availability of the functional product as without efficient and effective support the availability of the functional product will be solely dependent upon the hardware reliability. The lower the availability of the functional product the less economically beneficial it is. The reliability of hardware has been studied extensively and methods developed have reached an advanced level of sophistication, see Andrews and Moss (2002) and Gnedenko and Ushakov (1995). The reliability of service support systems has received considerably less attention. Li and Thompson (2009) adopted simulation techniques to assess the service reliability in the context of functional products.

The availability of the Functional Product is dependent on the reliability of both the hardware and the service support system. However, hardware reliability and the service support system, influence each other and are not independent. For example, increasing the reliability of the service support system affects the hardware reliability in two main ways, it reduces the number of system failures in systems with

redundancy by restoring subsystems before higher level failures can occur and it reduces the time that the system spends in the failed state through faster restoration.

Since the reliability of the hardware directly determines when and how often corrective maintenance is carried out, as well as being a major factor in determining preventative maintenance schedules, it affects the variation in demand for resources within the service support system. High demand and periods of peaking demand for resources caused by hardware unreliability can lead to increased lead times and thus service support system unreliability as maintenance procedures take longer to complete as they wait for required resources. In turn, the increased service support system unreliability can lead to increasing unreliability of the hardware as described previously, creating a positive feedback loop.

Despite the benefits of a Functional product, in order to be successful, its design must be such that it can achieve the desired availability at a satisfactory cost over its lifetime. This is a problem for potential suppliers as the only available method for modelling service support system reliability and cost, by Li and Thompson (2009), has some limitations. For example, it does not include some important factors that influence maintenance procedure reliability such as waiting times for common resources between tasks. Neither does it provide a mechanism for integration between the models of a Functional product’s hardware and service support system components that would allow the interactions between them to be modelled. In this work an improved model of the service support system, consisting of maintenance procedures and resource provisions, is developed that is able to interact with the system hardware model. This allows the reliability and cost of the service support system to be predicted, and, when later integrated with the hardware model, will enable the availability and cost of a particular Functional product design to be predicted.

2. MODEL DEVELOPMENT.

Due to the interactions discussed in the previous section, the system hardware and the maintenance procedures and resource logistics within its service support system must all be modelled as an integrated system. Each of these components has complex features that make simulation, rather than analytical techniques, the optimum method to be used for the modelling. In addition, simulation has the advantage that it can be used to produce a wide range of statistical results that are useful in the analysis of a Functional product design. Some of the statistics that can be produced include distributions of maintenance times, maintenance costs and expected resource utilisation. Petri nets were chosen to model the state of both the hardware and service support systems during simulation as they are well suited to modelling these types of dynamic system where state changes occur at discrete intervals and they also allow the interactions between the separate components to be modelled in a simple manner. . In simulation, many trials are carried out where each trial consists of generating random numbers to sample from the probability distributions present in the model and generate a representative outcome for the system. The outcomes from all trials are then used to generate statistics that are representative of the system.

Petri nets are a graph based tool that can be used to model the dynamics of many types of system, see Schneeweis (1999). Specifically, a Petri net is a directed bipartite graph in which each node represents either a transition or place, shown in diagrams as a bar or hollow circle respectively. Directed arcs linking places to transitions are known as inputs and those connecting transitions to places are known as outputs. In addition, multiple input or output arcs can link the same place and same transition, with the number of arcs known as the multiplicity, often represented as a single arc with a backslash through it and a positive integer denoting the multiplicity. Places may contain 0 or more tokens, represented by filled circles, and it is the distribution of tokens through the net, known as the net marking, that determines the state of the system. Each transition is associated and labelled with a time delay which may be fixed or determined from a distribution. When the number of tokens in a place matches or exceeds the number of input arcs, the transition is enabled and will fire once it has remained enabled for the duration of its associated delay, in which case the tokens are consumed from the input places, and deposited in the output places - thus altering the marking of the net and therefore the state of the system. The number of tokens consumed from the input place is equal to the number of input arcs and the number of tokens deposited in the output place is equal to the number of output arcs. If the marking of the net changes and disables a previously enabled transition, then that transition and its delay duration are reset. Only one transition can occur at any instant of time, regardless of the number of transitions that are enabled. An example of a transition, showing the before and after net markings, is shown in Figure 1.

Fig. 1. Petri net transition

2.1 Service Support System Model.

The maintenance of hardware within a service support system depends on maintenance procedures, resources and logistics, and scheduling.

In this work a maintenance procedure is represented by a special type of directed graph named an MP Graph. The graph consists of a source, and a terminal, node, indicating the start and end of the procedure respectively, and intermediate nodes representing tasks. Each task node can be labelled with its input resources, the process carried out and its output resources. The parent nodes of a task node represent its prerequisite tasks whilst its child nodes represent tasks for which it is a prerequisite. Prerequisite tasks that form a prerequisite set for a node are shown by edges that connect to the node at the same point. Each task node can also have two child node sets, where one set is for its failure and the other is for its success. Cycles in the graph show where rework can occur due to the failure of a task. An example of a MP Graph representation of a maintenance procedure is shown in Figure 2, where task details are omitted and only the task numbers are shown.

Fig. 2. An example of a MP graph representation of a maintenance procedure.

Task 4 has two prerequisite sets, both task 1 and 2 or task 3 must complete for it to be ready to start. Task 5 has two child node sets, if it completes successfully then it links to the end node, if it fails then it links back to task 3 as shown by the

dotted line in the figure. Hence failure of task 5 causes task 3 and 5 to be performed again.

The completion time for a task, the time duration from when it starts after the completion of a prerequisite set and allocation of resources to when it finishes, is modelled through a distribution function whose output is positive real values. Each task is also associated with a reliability value that gives the probability that it completes successfully on each execution.

In order to perform the simulations the MP Graphs are converted into their Petri net representations prior to analysis. Figures 3a,b show the MP graph and Petri net representation respectively for a simple maintenance procedure consisting of one task that must be repeated if it fails. In Fig. 3b places labelled S and E are the ‘Initiate Maintenance Procedure’ and ‘Maintenance Procedure Complete’ places respectively. The places labelled AR and RA are the ‘Awaiting Resources’ and ‘Resources Allocated’ places respectively.

Fig 3a. MP graph for simple maintenance procedure

Fig. 3b Petri Net representation for the simple maintenance procedure.

When a token is deposited in the “Initiate Maintenance Procedure” place in Fig. 3b, the transition fires after a delay S_{MP} , corresponding to the time delay for starting the maintenance procedure, and deposits a token in the prerequisite set place of the Petri net representations of the task node in the MP Graph. For the simple example shown this causes an instantaneous transition to occur that sends a request for the allocation of the resources required to complete the task to the Resources Model, and deposits a token in the “Awaiting Resources” place. Once the Resources Model has allocated the required resources and deposits a token in the “Resources Allocated” place, the final transition fires after delay C_T . C_T represents the completion time for

the task and is sampled from a probability distribution. This final transition is a Dual Output transition, a new type of transition introduced here, and deposits tokens in either the prerequisite place, for the end node on success, or to repeat the task on failure (dashed arc), and also notifies the Resources Model that the task has completed. Whether the task completes successfully or not is chosen randomly based on the reliability of the task. If the task completes successfully the final transition fires after a delay E_{MP} , representing any delay in completing the maintenance procedure once the task is complete. The total time taken to complete the maintenance procedure is the time taken from the moment that a token is deposited in the “Initiate Maintenance Procedure” place to the moment a token is deposited in the “Maintenance Procedure Complete” place.

More complex maintenance procedures can be modelled in the same way. As seen in the above example, the Petri net for a maintenance procedure task notifies the resource model that it requires the allocation of resources once it is ready to start, the resources and logistics models then processes this request and once it can allocate the appropriate resources, deposits a token in the “Resource allocation” place of the maintenance procedure task Petri net. This interaction between the maintenance procedure and resource models, along with the processing and data flow within the resource model, is described by the Gane-Sarson, Gane and Sarson (1979), diagram shown in Fig. 4. Due to the huge variety and complexity of supply chains, a generic methodology for the internal processing of the resource model is not possible. Two simple models for the “Find Resources” process shown in Fig. 4 have been implemented.

Fig.4 Gane-Sarson diagram for the resource model

To associate a maintenance procedure with the repair of a particular subsystem or component it is necessary to integrate the maintenance procedure Petri nets into the hardware reliability Petri nets. This is achieved through the use of maintenance integration Petri nets which link the ‘system down’ place in the hardware Petri net to the maintenance procedure Petri net, like those shown in Figs 3a,b. An example of a maintenance integration Petri net is shown in Fig. 5. Initially, before maintenance has begun, a token is present in the “Maintenance Waiting” place of the Petri net, whilst all others are empty. Depositing a token in the “Start Scheduled Maintenance” place then causes the preventative maintenance transition to fire, depositing a token in the “Initiate Maintenance Procedure” place of the maintenance procedure Petri net (not shown in the figure). Alternatively, a token can be deposited in the “Inspect System” place which causes the corrective maintenance transition to fire and deposit a token in the “Initiate Maintenance Procedure” place of the maintenance procedure Petri net if a token is present in the “System Down” place within the inspection period, I_p . The “Inspect System” place is removed if failures are revealed, causing corrective maintenance to begin immediately upon failure of the item.

With both corrective and preventative maintenance, once the maintenance procedure is complete and a token is present in its “Maintenance Procedure Complete” place, another transition fires and deposits a token in the “Maintenance Complete Place”. This causes the final transition in the generic maintenance integration Petri net to fire, restoring the components within the item (e.g. subsystem) to the “as new” condition and depositing a token in the “Maintenance Waiting” place so that further maintenance can be initiated in the future.

Fig. 5. Example of a maintenance integration Petri net.

The service support system influences the maintenance schedule by deciding when and how often a token is deposited in the “Inspect System” and “Start Scheduled Maintenance” Petri net places.

The final component of the service support system reliability model is the Event Scheduler. Each model generates events that occur at some point in the future of the current simulation time and these must be synchronised so that they

are processed in the correct order. For example, the hardware model generates hardware failure events, the maintenance procedure model generates task completion events and the resources model generates resource allocation events. The Event Scheduler is responsible for tracking the simulation time and processing all the events generated, executing them in the correct order.

The model described above has been validated by applying it to standard maintenance procedures. An example of this is given below.

3. EXAMPLE

As an example of an application of the methodology described a service support system consisting of a single maintenance procedure, changing of a car wheel, has been considered. The MP graph representation of the procedure is shown in Figure 6 where the task numbers are defined in Table 1. Also included in Table 1 are the input and output resources for each task.

Fig 6. MP graph representation of the wheel change maintenance procedure.

Task No.	Description	Input resources*	Output resources*
1	Apply the handbrake		
2	Get the spare wheel		Spare wheel
3	Get the Wheelbrace		Wheelbrace
4	Get the jack		Jack
5	Remove wheel trim		Wheel trim
6	Loosen the wheel bolts	Wheelbrace	Wheelbrace
7	Position jack at the jacking point	Jack	
8	Raise the jack		
9	Remove the wheel bolts	Wheelbrace	Wheelbrace, Wheel bolts.
10	Remove the wheel		Replaced wheel
11	Fit the spare wheel	Spare wheel	
12	Replace bolts	Wheel bolts	
13	Tighten bolts	Wheelbrace	Wheelbrace
14	Lower and remove		Jack

	the jack		
15	Tighten the bolts fully	Wheelbrace	Wheelbrace
16	Refit the wheel trim	Wheel trim	
17	Put away the Wheelbrace	Wheelbrace	
18	Put away the replaced wheel	Replaced wheel	
19	Remove damaged bolts	Wheelbrace	Wheelbrace

* All tasks also have a person as both an input and output resource.

Table 1. Task number description

In the model normal distributions were used to model task completion times with values for the mean and standard deviation of the distributions assumed. The assumption was also made that task 13 can fail if the wheel bolts are damaged. In that case new wheel bolts would be ordered from a supplier with a lead time assumed. The system was analysed using the techniques described here for various situations. Table 2 details some of the situations along with the mean, maximum and minimum completion times for each scenario

Scenario	Number of people	Reliability of task 13 (table 1)	Lead time for supply of new bolts (seconds)	Mean completion time
1	1	0.9	100	617
2	2	0.9	100	531
3	5	0.9	100	503
4	1	0.7	100	719
5	1	0.9	300	639

An example of the results is shown in Figure 7a,b,c where the distribution of the completion time for the maintenance procedure is shown when the number of people undertaking the procedure is a)one, b) two and c) five. The only change to the modelling between the three situations is therefore in the initial number of people available in the resources model that determines the allocation of resources to maintenance tasks as shown by Fig. 3b.

Fig. 7a

In Fig. 7a, since every task requires a person to carry it out and only one person is available, the tasks are carried out sequentially. The MP Graph shown in Fig. 6 is general and allows for some concurrency. Therefore tasks that would otherwise start whilst another task is being performed are awaiting the allocation of the person occupied by that task. The cluster on the left of the figures corresponds to the completion of the maintenance procedure when task 13 completes successfully at the first attempt, whilst the rightmost corresponds to a failure of task 13 on the first attempt and success on the second. The gap between the two clusters is due to the corrective actions, delay in obtaining new bolts and repeat of task 13.

Fig. 7b

Fig. 7b shows the distribution of completion times for the maintenance procedure for the case when 2 people are available. In this scenario, with an extra person available, some of the tasks can be carried out in parallel. As shown in Fig. 7b, this leads to the clusters of likely completion times to be shifted to lower times with the shape of the distribution largely unchanged.

Fig. 7c

Fig 7. Distribution of the completion time for the maintenance procedure

In the situation when 5 people are available, results in the distribution of most likely completion times are shown in Fig. 7c. This shows a further shift in the distribution to lower times, however compared to the shift between the cases of one and two people available, the difference is far smaller. The reason for the diminishing reduction in completion time as each extra person is added to the available resources is that the number of people is no longer the limiting factor for the number of tasks that can be performed simultaneously and therefore the extra resources are not fully utilised. The limiting factors are that certain tasks cannot be carried out simultaneously due to the scheduling in the maintenance procedure shown by its MP Graph, Fig. 6, and the limited availability of other common resources.

4. CONCLUSIONS

The ability to model the reliability of a service support system is important for the development of functional products. To date work in this area has been limited, with a basic simulation model developed by Li and Thompson (2009). In the work presented here new methods for the representation and modelling of the reliability of a service support system through Monte Carlo simulation and Petri nets have been demonstrated. These techniques are very versatile and allow for many scenarios to be considered as well as the integration of the service support system and hardware models. This is an advance on the earlier model. The techniques adopted have been found to be an efficient and effective means of modelling the system. The maintenance procedure modelling has been verified through application to a number of different maintenance procedures, including the wheel change shown in this paper. Further work is necessary to extend the modelling to maintenance procedures with more complex sequencing constraints, such as maintenance procedures that do not start until the resources for all its tasks are available and task schedules that depend on the intelligent sharing of common resources between different maintenance procedures that are carried out simultaneously.

REFERENCES

- T. Alonso-Rasgado, G. Thompson and B. Elfström, "The design of functional (total care) products," *J. Eng. Des.*, vol. 15, pp. 515, 2004
- J. D. Andrews and T. R. Moss, *Reliability and Risk Assessment*. 2nd ed. Professional Engineering Publishing Limited, 2002,
- C. Gane and T. Sarson, *Structured Systems Analysis: Tools and Techniques*. Prentice Hall PTR, 1979, pp. 288.
- B. Gnedenko and I. A. Ushakov, *Probabilistic Reliability Engineering*. John Wiley & Sons, 1995, pp. 517.
- J. Li and G. Thompson, "Simulation-based Reliability Assessment of Services in the Context of Functional Products," *J. Safety and Reliab. Soc.*, Vol. 29, pp47, 2009.
- W. G. Schneeweiss, *Petri Nets for Reliability Modeling*. LiLoLe-Verlag GmbH, 1999,