Simulation and Calculation of Reliability Performance and Maintenance Costs

Per-Erik Hagmark, PhD, Tampere University of Technology

Seppo Virtanen, PhD, Tampere University of Technology

Key Words: simulation, reliability performance, maintenance costs

SUMMARY & CONCLUSION

Our model for the simulation and calculation of reliability performance and maintenance costs is based on a generalized fault tree approach, where the TOP represents the product to be designed. The fault tree consists of entities, which essentially affect the failure tendency and the repair time of the product. The relations between entities are modeled by two mechanisms. The "gates" determine the partly logical and partly stochastic propagation of faults (primary states). The "strategies" define other relations between TOP and the deepest entities. A consequence of the strategies is that two types of "waiting" (secondary states) can occur.

The model parameters have been selected to make the assessment by the designer as easy and reliable as possible. Our model provides 10 different methods to design the cumulative distribution function for the average number of failures and for the repair time. The designed repair time often includes delays with external causes. The developed software (RAMoptim) also supports the separate addition of delays. The lack of repair staff can be one example. Another example can be the lack of spare parts, which can be assessed with software (StockOptim), also developed in the research project.

When using the developed method with corresponding software, the designer can determine at an early stage of the design to which level of reliability performance and maintenance costs can be achieved by using the selected design solution and maintenance strategies. The method can also be used to import expertise into the design process from areas that strongly affect the success of that process, namely the manufacturing, testing, operation, and maintenance of the product. If the defined requirements have not been achieved, the designer must go back to the drawing table to consider other solutions for achieving the requirements (Fig. 1). The RAMoptim software also includes computer supported methods developed to quantify the effect of preventive maintenance (PM) on a part's failure tendency. With the help of this method, reliability performance, repair and preventive maintenance costs can be optimized. The condition monitoring resources are included in preventive maintenance resources as well.

The applicability of the developed methods and software has been tested in the companies participating in the research project. Based on the experience, and with the help of the methods and corresponding software, it is possible to identify those problem areas during the design stage which can delay the product development and/or reduce the safety and reliability. Ramentor Oy (www.ramentor.com) is responsible for commercializing, marketing and supplying technical support of the developed computer software

The application of the methods has guided the companies to transfer their resources from failure repairs to prevent them during the design stage. Correspondingly, the application of the method has forced companies to improve their failure and preventive maintenance reporting systems. In addition, the companies have noticed that the engineers need more knowledge in the area of probabilistic approach in reliability and maintainability engineering.

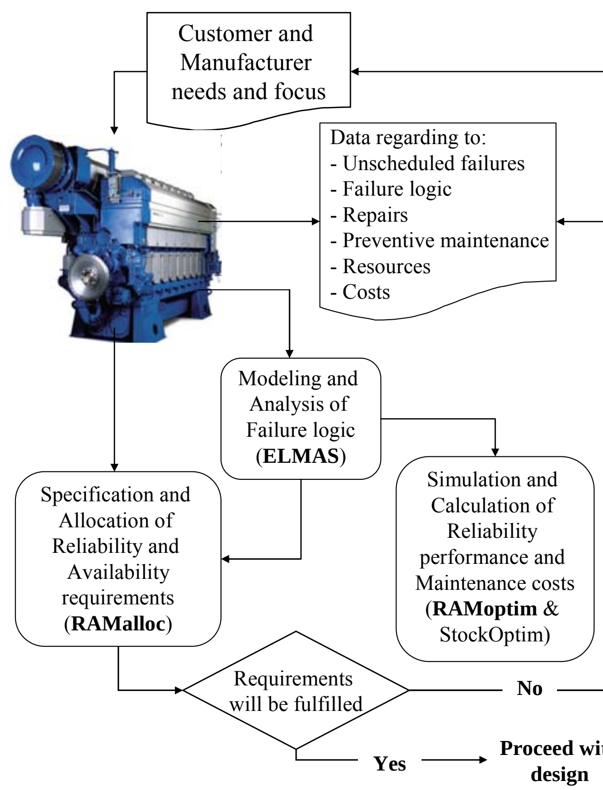

Figure 1 - Probabilistic approach to defining Reliability and Availability requirements to the product and to assess that a proposed design solution fulfils the numeric requirements set for its Reliability performance and Maintenance costs

1 INTRODUCTION

This paper discusses the method for simulation and calculation of reliability performance and maintenance costs of a design entity. The general term "entity" can stand for function, system, equipment, mechanism, part, etc.

The method is one of the main results from the research project, which lasted about nine years and was carried out by Tampere University of Technology. Since 1996 eleven Finnish companies have participated in the research project, whose objective was to develop computer supported probability based methods for the development of the equipment's and systems' reliability and safety. The participating companies are both manufacturers and users of the equipment, in metal, energy, process and electronics industries. Their products and systems have to correspond to high safety and reliability demands.

2 FAILURE AND OPERATION STRUCTURES

2.1 Stochastic Failure Logic

The product under design is represented by a generalized fault tree, which describes how failures can propagate from one *entity* (part) of the product to another /1/. The (primary) *state* of an entity is either 1, failed, or 0, non-failed. An entity is here identified as either a *gate* or a *basic part*, shortly *BP*. The state of a gate depends causally on the states of certain entities, the inputs of the gate. This mechanism is partly logical and partly stochastic. It is characterized by giving the data vector ($ID, k, m, P, \pm I_1, \pm I_2 \ldots \pm I_n$), where ID is the number of the gate, and I_i are the numbers of the input entities, where a minus–sign denotes that the input is first negated. Now, if $k \leq$ the sum of {0,1}-inputs $\leq m$, then the gate adopts the state 1 with probability P, otherwise its state is 0.

The structure matrix (diagram) of a fault tree contains the data vectors of the gates as columns. The simple example (1), which will be followed up in subsequent sections, has the gates 3, 5, 6 (TOP), and the basic parts 10, 8, 9, 15.

$$\begin{bmatrix} ID & 6 & 3 & 5 \\ k & 2 & 1 & 1 \\ m & 2 & 2 & 1 \\ P & 0.9 & 0.8 & 1.0 \\ I_1 & 3 & 8 & 15 \\ I_2 & 5 & 9 & 10 \end{bmatrix} \tag{1}$$

2.2 Operation Strategies

A *BP* has no inputs and its primary state (0 or 1) is generated independently of other entities. A *BP* should therefore preferably be a physical-technical entity, for which it is possible to assess repair time and failure tendency according to Sec. 3 below. Every time the state of some *BP* changes, the mechanism defined above leads to logical states for all entities including TOP, and this generation is performed in an order not conflicting causality.

The failure logic determines the primary states in the fault tree but certain additional interrelations or bounds can yet exist. Many of these relations can be modeled by setting for each *BP* some combination of the following deterministic operation strategies:

a=1: This BP cannot be repaired if TOP is running. Otherwise a=0
b=1: This BP is not running if TOP is not running. Otherwise b=0
c=1: TOP will not be started if this BP is still failed. Otherwise c=0

The *b*-strategy can also be used for gates. Note that any binary triple *a,b,c* is in principle possible for a *BP*. The choice *a*=1 (valid), *c*=0 (negated) can, however, lead to non-practical behavior, where the repair of a *BP* is interrupted before it is ready, and is continued later.

2.3 Waiting States

The operation strategies generally imply the existence of two new states, the waiting states (0.5 and 1.5). Thus the complete set of possible states for entities is:

0 *Non-failed and running*
0.5 *Non-failed but start is denied*
1 *Failed and available for repair*
1.5 *Failed but repair is denied (only for BPs)*

For example, when a repair of a *BP* is finished, the state changes from 1 to 0 or 0.5, and when the *BP* fails, its state changes from 0 to 1 or 1.5. In general, a change in the state of an entity is always caused by the appearance or disappearance of failure somewhere in the fault tree.

Note that scheduled or pre-planned stops (of any kind) are not waiting states. These can, as well as the corresponding unavailability, be taken into account beforehand or afterwards.

3 INPUT DATA AND SIMULATION

3.1 Repair time

A *BP*'s repair time is the length of a period in state 1. Our software offers 10 methods to build the required cumulative distribution function (Cdf). Modifications of standard distributions, mixtures, splines and other tailor-made constructions have been employed to guarantee sufficient flexibility. Depending on the sort of available input data, various combinations of means, deviations, medians, quantiles, minimums, maximums, censoring, weights, experts´ competence, etc., can be used.

The designed repair time includes often delays having outer causes. The software supports also separate adding of delays. The lack of repair staff can be of this kind. Further, the lack of spare parts is of this kind, and the corresponding delay can in fact be assessed with another module of the software (StockOptim), also developed in the research project /2/. On the other hand, delays caused by operation strategies (states 1.5 and 0.5) are of course not of this kind.

3.2 *Failure tendency and failure profile*

Each *BP* has usually a natural own "unit of usage" (uu) for the measuring of operative usage. The failure tendency of a *BP* is defined with a non-decreasing, possibly non-linear function, whose value $v = \Lambda(x)$, $\Lambda(0) = 0$, expresses the cumulative average number v of failures in the usage period $0 \ldots x$ (uu). Depending on the nature of available data, our model provides 10 methods for the design of $\Lambda(x)$. Supported types of data are the number of failures, failure rates, reliability, availability, censoring, life distributions, etc. One can also experiment with changes in the length of service period for exchangeable *BP*s.

Next the usage profile can be assessed. This is also a non-decreasing function, whose value $x = \Psi(t)$, $\Psi(0) = 0$, is the average amount of usage achieved by a *BP* in the active age period $0 \ldots t$ (tu). Note that t cumulates only the *BP*'s own operative time (state = 0), so $\Psi(t)$ is not directly a "plan for usage" during the design period. Before the simulation (Sec. 3.4), it is not even known how the operative time of the BP grows during the design period.

Now the average number of failures occurring in the age period $0 \ldots t$ (tu) is the compound of the usage profile and the failure tendency: $v = \Lambda(\Psi(t))$. This is the cumulative *ROCOF-function* or failure profile, which is to be used in the simulation. The generation of the failure instants of a *BP* during simulation will follow the *NHPP*-process, whose intensity is the derivative of the failure profile.

3.3 *Effects caused by failures and stops*

In many cases failure-caused stops have directly or indirectly some effect on *BP*s subsequent behavior. It is assumed that these effects can be interpreted as changes in the age of the *BP* or in its probability to fail. Our model offers for example the following parameters:

$L \geq 0$	*Age of the BP at start*
$Y \geq 0$	*Age correction factor after repair (NewAge = Y·OldAge)*
P	*Failure probability immediately after repair*

The parameter L handles for example the following situation: The failure profile was perhaps constructed for a new *BP*, but the *BP* to be used in the product is L (tu) old.

3.4 *Continuous simulation and the logbook*

At this stage continuous simulation will be performed. In the beginning all *BP*s are working (state 0), and random time to failure is generated for each *BP*. When the first *BP* fails, the state of the fault tree is generated up to TOP according to Sec.2. If TOP is still working (0) and a=1, the failed *BP* takes the "waiting-for-repair" state 1.5, otherwise (TOP failed or a=0) the failed *BP* takes the state 1, and repair time is generated. Besides, if TOP failed, then those working (non-failed) *BP*, whose b=1, must go into the "waiting-for-start" state (0.5). Etc.

This goes on until the end of the design period, i.e., until the running time of TOP reaches a predefined value, T (tu). A sufficiently large number of equivalent simulation rounds follow. All this is registered in a final simulation document, the logbook describing events, time instants, TOP-age instants, states, duration of state combinations, etc.

A direct and detailed study of the logbook is the designer's best preliminary test for how successfully the model and the input data match the desired behavior. The logbook in Table 1 is the raw data from the simulation of the fault tree example (1). The first row shows the number of simulations (N = 1000), design period, i.e., TOP-age interval (T = 25000), TOP (ID = 6), basic parts (ID = 10, 8, 9, 15), and other gates (ID = 5, 3). The following rows describe in chronological order and completely the behavior of the built model: cumulative total time, active TOP-age, state combination for the entities, and time duration of this combination.

Ev =

	0	1	2	3	4	5	6	7	8	9	10
0	1000	25000	6	10	8	9	15	3	5	6	0
1	0	0	0	0	0	0	0	0	0	0	269.9
2	269.9	269.9	1	1	0.5	0.5	0.5	1	1	1	2.8
3	272.7	269.9	0	0	0	0	0	0	0	0	172.3
4	445	442.2	1	0	0.5	0.5	1	1	1	1	10.1
5	455	442.2	0	0	0	0	0	1	0	0	867
6	1322	1309.2	0	1.5	0	0	0	0	1	0	850
7	2172	2159.2	0	1.5	0	0	1	1	0	0	4.2
8	2176.2	2163.4	1	1	0.5	0.5	0.5	1	1	1	5
9	2181.2	2163.4	0	0	0	0	0	1	0	0	128.7
10	2310	2292.1	1	1	0.5	0.5	0.5	1	1	1	4.4
11	2314.4	2292.1	0	0	0	0	0	1	0	0	585.2
12	2899.6	2877.3	0	0	0	1	0	0	0	0	9.4
13	2909	2886.7	0	0	0	0	0	1	0	0	106.8
14	3015.7	2993.5	0	0	0	1	0	0	0	0	7
15	3022.8	3000.5	0	0	0	0	0	1	0	0	168.1
16	3190.9	3168.6	0	0	0	1	0	0	0	0	6.4
17	3197.3	3175.1	0	0	0	0	0	1	0	0	630.3
18	3827.6	3805.3	0	0	1.5	0	0	1	0	0	2757.7

Table 1 – The first page of the logbook

The first general extract from the logbook depicts how much time the entities spend in 0-state. Table 1 yields the numerical result:

$$\begin{bmatrix} 6 & 10 & 8 & 9 & 15 \\ 25000 & 21591 & 16551 & 24965 & 24960 \end{bmatrix} \qquad (2)$$

A question could arise. Why does *BP*=8 spend so much time in waiting states? This can perhaps originate from the selected operation strategies (Sec.2.2-3).

4 RELIABILITY CALCULATION

4.1 *Main results for the product (TOP)*

Various measures for the TOP's reliability performance can be extracted from the logbook. In the following we present three main functions (curves) which are combined from the logbook (Table 1).

The length of an unbroken non-operative time period is a random variable, here called single downtime or "downtime period". A TOP-downtime period consists of states 0.5 or 1 (TOP never adopts the state 1.5, Sec.2.3). The associated Cdf for TOP=6 in fault tree (1) is depicted in Fig.2.

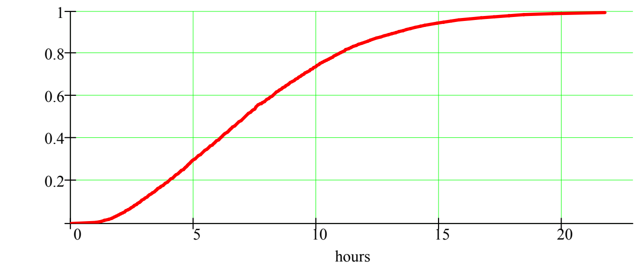

Figure 2 - Single downtime; mean 7.6, deviation 4.1

The *failure profile* is consequently the cumulative number of downtime periods during the design period. See Fig.3 for TOP=6! (The dotted lines are the 5% and 95% quantile profiles.)

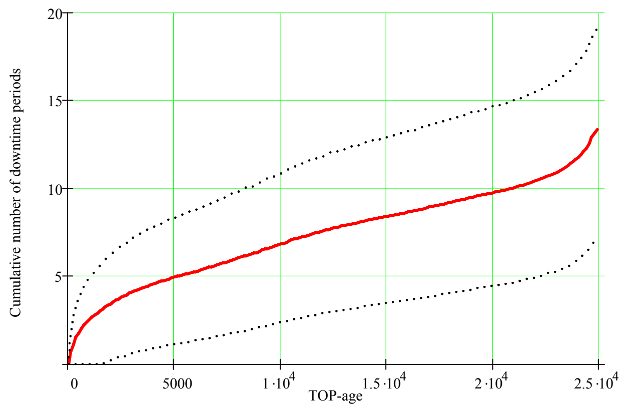

Figure 3 - Failure profile; mean 13.3, deviation 3.1

A useful combination of downtime and failure profile is given by a (smoothed) availability curve, the points of which are harmonic averages. See Fig.4 for TOP=6!

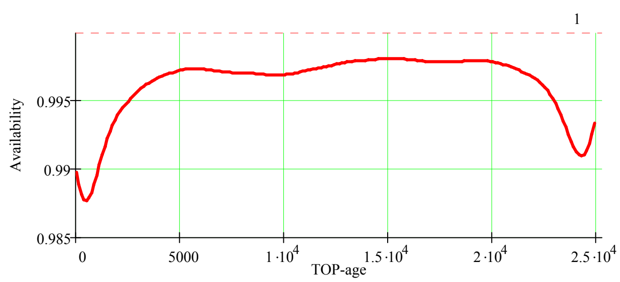

Figure 4 - Point wise availability; mean 0.9959

Remark. The functions (Fig. 2 – 4) should be compared to possibly existing reliability and availability requirements, assessed perhaps by using the allocation method, also developed in the research project /3/. If the TOP is (also) a *BP* in a higher fault tree, its downtime will simply be called repair time, since downtime is always caused by failures, directly or indirectly. In that case, these functions defined availability required, according to the definitions sections 3.1-2.

4.2 Reliability performance figures for the product

Further outputs from the calculation are given by the following figures for TOP in the design period 0...*T*.

Duration and number of downtime and corresponding deviations
Duration and number of repairs and corresponding deviations
Duration and number of waiting and corresponding deviations
Duration of single downtime period and corresponding deviation
Duration of single operation period (MTTF) (0) and corresponding deviation
Duration of single repair (MTTR) (1) and corresponding deviation
Duration of single waiting (0.5) and corresponding deviation
Mean time to first failure (MTTFF) and corresponding deviation
Maximum (95%) duration of single repair (1)
Maximum (95%) duration of single downtime (0.5 or 1)
Failure probability within desired interval $t_1 ... t_2$
Average unavailability (caused by failures)

There are random variables behind the figures in the list above. More detailed and applicable information from the calculation is given by the cumulative distributions functions Fig.5 and Fig. 6 show two examples!

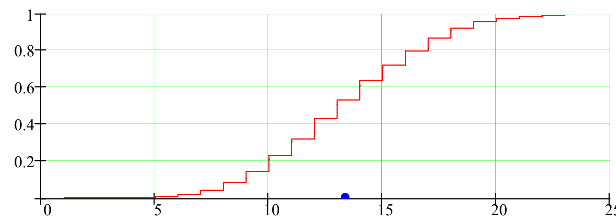

Figure 5 – Number of downtime periods; mean 13.3, deviation 3.7

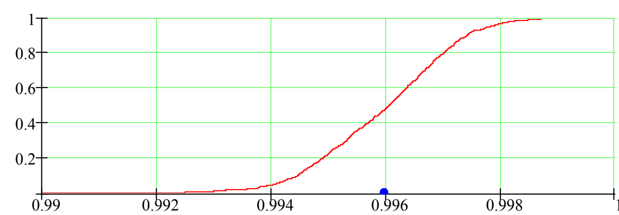

Figure 6 - Total availability; mean 0.9959, deviation 0.0012

4.3 Entities involved in TOP-downtime

The role of a certain entity in TOP-downtime can be considered from various points of view. Our method and software provide the probability that an entity takes part in a downtime period, how long does the situation last, availability for each entity, etc. Here we list the following figures:

P_1 *Time share of an entity's failure state of TOP's total down time*
P_2 *Probability that an entity is already failed or fails simultaneously when TOP goes to an downtime state*

The numerical version (3) is the result of example (1):

$$\begin{bmatrix} ID & 10 & 8 & 9 & 15 & 3 & 5 \\ P_1 & 0.2533 & 0.2922 & 0.0117 & 0.6405 & 0.9774 & 0.8936 \\ P_2 & 0.4712 & 0.2762 & 0.0127 & 0.5288 & 1 & 1 \end{bmatrix} \qquad (3)$$

5 COST AND RESOURCE CALCULATION

5.1 Additional inputs concerning failures

The logbook (Table 1) constitutes a list of all stochastic events caused by failures, and thereby it is the raw material for the calculation of failure-caused costs and resources. The designer is first asked for the following additional input data for each entity (ID):

Lf loss caused by (one) downtime (€)
Ln loss caused by downtime (€/tu)
d downtime (tu) without costs
R number of persons needed for repair (average)
G time independent repair cost (€)
H cost per repair time unit (€/tu)

Remark. The possibility to assess gates (including TOP) independently of their input entities offers an additional degree of freedom. For example, a gate failure can cause an extra loss, in addition to the loss caused by failed input entities.

5.2 Costs and resources due to failures

The first calculative result consists of the following averages for each entity, and the list (4) presents numerical results for fault tree (1). This list corresponds to the rows 0 – 10 in (4).

0 ID-number for entity
1 Running time, 0 (tu)
2 Waiting for start, 0.5 (tu)
3 Failed and available for repair, 1 (tu)
4 Failed but waiting for repair, 1.5 (tu)
5 Duration of downtime, 0.5, 1, 1.5 (tu)
6 Number of failures
7 Time independent repair costs (€)
8 Time-dependent repair cost (€/tu)
9 Loss caused by failure (€)
10 Loss caused by downtime (€/tu)
Last column = Total (sum of row)

$$\begin{bmatrix} 0 & 6 & 10 & 8 & 9 & 15 & Total \\ 1 & 25000 & 21591 & 16551 & 24965 & 24960 & \\ 2 & 10.9 & 0 & 72.0 & 100.5 & 36.6 & \\ 3 & 90.9 & 25.8 & 29.7 & 36.5 & 105.2 & \\ 4 & 0 & 3484.5 & 8448.5 & 0 & 0 & \\ 5 & 101.7 & 3510.2 & 8550.2 & 137.1 & 141.7 & \\ 6 & 13.4 & 6.3 & 3.8 & 4.9 & 11.1 & \\ 7 & 1134.5 & 946.9 & 759.6 & 1477.5 & 3880.1 & 8198.6 \\ 8 & 7267.8 & 1545.8 & 3567.0 & 6572.3 & 12619.2 & 31572.1 \\ 9 & 13347.0 & 1578.3 & 379.8 & 1970.0 & 1662.9 & 18938.0 \\ 10 & 10172.6 & & & & & 10172.6 \end{bmatrix} \quad (4)$$

Note that the results given in (4) hold for TOP as an entity (the first column)! The following results are for the entire product. They show how long time a certain number of persons is needed (simultaneously). The numeric result is given in (5) where

Rn number of persons
Pr time needed simultaneously for the number of persons
PWR total work time (mean) of persons
Rr average number of persons needed simultaneously

$$\begin{bmatrix} Rn & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & PWR & Rr \\ Pr & 0 & 68.0 & 89.3 & 7.3 & 11.7 & 0 & 0.61 & 0.02 & 496.0 & 2.8 \end{bmatrix} \quad (5)$$

For example, the total time for exactly 3 persons was 89.3 (tu). From (5) can be seen how much staff is needed on average (Rr = 2.8), and how much work time was totally spent for repair (PWR = 496 tu). (A practical conclusion could perhaps be that a repair staff of 3 persons would not cause too much delay.) Our model also provides results that complete the new of product (TOP) as BP in a higher level of fault tree (compare Sec. 4.1), for example the average prices for loss and repair, and average number of persons needed simultaneously (Rr).

5.3 Documentation of scheduled procedures

When failure tendency was designed for the *BP*s of the product (Sec. 3.1-2), the designer used of course assumptions and background knowledge about *corresponding scheduled procedures*, for example, concerning preventive maintenance and condition monitoring. The assumed non-casual events need now to be documented in detail for further cost and resource calculations.

The method allows the designer to associate different scheduled procedures, *SP* = 1, 2…, to each *BP*. For example, if the *BP* is a car, then *SP* = 1 can denote “change of oils”, *SP* = 2 “large service package for a fairly new car”, *SP* = 3 “large service package for an old car”, etc. Then the designer assesses the following parameters, for each *BP* and each selected *SP*:

μ *average duration (tu)*
σ *deviation of duration (tu)*
R *persons needed simultaneously (average)*
G *time independent costs (€)*
H *time dependent costs (€/tu)*

Then the entire schedule for the procedures is built up, event by event, according to the format illustrated in Table 2. Each row carries the information of procedures done simultaneously to certain entities. The moment for these procedures is defined by the TOP-age in the first column. The second column shows the sign 0 or 1 according to whether TOP can or cannot be running during the procedures. The format of the following columns is *BP.SP*, which shows that the procedure *SP* of the basic part *BP* is performed. For example, 8.1 means that *BP*=8 will be addressed by its *SP*= 1.

S^T =

	0	1	2	3	4	5	6	7	8	9
1	1000	1	8.1	9.1	10.1	15.1	0	0	0	0
2	5000	0	8.1	9.1	10.1	15.1	0	0	0	0
3	8000	0	8.3	10.3	0	0	0	0	0	0
4	10000	1	8.1	9.1	10.1	15.1	8.2	9.2	10.2	15.2
5	16000	1	8.3	10.3	9.3	0	0	0	0	0
6	20000	1	8.1	9.1	10.1	15.1	8.2	9.2	10.2	15.2

Table 2 – An entire schedule for the procedures

Note that when planning an alternative schedule, a *BP*'s own usage is essential rather than the TOP-age (left column). The additional information needed comes from the failure profile (Sec.3.2 and 2). If a *BP* spends much time waiting, it can perhaps be served more rarely.

5.4 Costs and resources due to scheduled procedures

The schedule (Table 2) is a frame for cost and resource accounting associated with constant events. Some of the results are given in (6), where:

GG Time independent SP-costs (€), mean
HH Time dependent cumulative SP-costs (€); mean, deviation
DurS Total duration of SPs (tu); mean, deviation
NonS Non-running time caused by SPs; mean, deviation
AvS Average unavailability caused by SPs

$$\begin{bmatrix} GG & HH & DurS & NonS & AvS \\ 4850 & 5600,\ 755 & 30.9,\ 5.9 & 22.1,\ 4.9 & 0.0009 \end{bmatrix} \quad (6)$$

The next results of calculation shows how long time (*PrS*) a certain number of persons (*RnS*) is needed simultaneously to carry out *SPs*. The numeric result is given in (7) where PWS is total person work time caused by SPs, and Rs is number of persons needed simultaneously, average.

$$\begin{bmatrix} RnS & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & PWS & Rs \\ PrS & 12.7 & 5.2 & 5.2 & 3.6 & 0.7 & 0.6 & 0.6 & 0.9 & 1.5 & 85 & 2.7 \end{bmatrix} \quad (7)$$

A practical conclusion could perhaps be that a maintenance staff of 3 persons would not cause too much delay. Table 3 presents the summary of the numeric results from the example we have introduced previously. The information is combined from (4), (5), (6) and (7).

Total availability	**0.9950**
Unavailability caused by failures	0.0041
Unavailability caused by SPs	0.0009
Total maintenance costs	**79331**
Costs caused by failures	68881
Loss caused by failure	18938
Loss caused by downtime	10173
Time independent repair costs	8199
Time dependent repair costs	31572
Costs caused by SPs	10450
Time independent SP-costs	4850
Time dependent SP-costs	5600
Required maintenance resources	
Total work time	**581**
Persons needed simultaneously	**2.8**
Caused by failures	
Total repair time	496
Persons needed simultaneously	2.8
Caused by SPs	
Total time for SPs	85
Persons needed simultaneously	2.7

Table 3 – Summary of reliability performance and maintenance costs

REFERENCES

1. Virtanen, S, Hagmark, P-E and Penttinen J-P. Modelling and Analysis of Causes and Consequences of Failures. *Annual Reliability and Maintainability Symposium* (RAMS). January 23 – 26, 2006. Newport Beach, CA, USA.
2. Hagmark, P-E & Pernu H. 2006. Risk evaluation of a spare part stock by stochastic simulation. *ESREL 2006 Conference, Safety and Reliability for Managing Risk*. 18 – 22 September 2006 – Estoril, Portugal.Fault
3. Virtanen, S & Hagmark, P-E. Specification and Allocation of Reliability and Availability Requirements. *Annual Reliability and Maintainability Symposium* (RAMS). January 23 – 26, 2006. Newport Beach, CA, USA.

BIOGRAPHIES

Seppo Virtanen, PhD
Institute of Machine Design and Operation
Tampere University of Technology
Korkeakoulunkatu 6, FI-33101 Tampere, Finland

e-mail: seppo.virtanen@tut.fi

Seppo Virtanen received his B.Sc., M.Sc. and PhD. degrees from Helsinki University of Technology, Finland. He is currently a Professor in the Machine Design and Operation Laboratory at the Tampere University of Technology. His research and teaching interest includes reliability and maintainability engineering and risk management within a product and system design process. Professor Virtanen has over 15 year's industry experience in the field of reliability engineering and maintenance, which includes three years in energy, pulp and paper industry in USA and two years offshore industry in Norway

Per-Erik Hagmark, PhD
Institute of Machine Design and Operation
Tampere University of Technology
Korkeakoulunkatu 6, FI-33101 Tampere, Finland

e-mail: per-erik.hagmark@tut.fi

Per-Erik Hagmark serves on the Institute of Machine Design and Operation at Tampere University of Technology, Finland. He studied mathematics, applied mathematics and theoretical physics at Helsinki University of Technology, and earned his doctoral degree in 1983 with a dissertation on the 2-series field (generalized Walsh functions) with applications. His early research interest and activity were around Cayley-Dickson algebras, Clifford algebras, fast orthogonal transforms, coding, and stochastic simulation. The last ten years consist mainly of research and model building in matrix experiments, reliability, maintainability, warranty, etc.