A RELIABILITY MODEL USING MARKOV CHAINS FOR UTILITY EVALUATION OF COMPUTER SYSTEMS ONBOARD SHIPS

Carsten Bøe, Tor Heimly and Tor-Christian Mathiesen

Det norske Veritas

Oslo, Norway

Abstract

Introduction of computers onboard ships to provide a high degree of automation necessitates calculation of computer system reliability to evaluate the utility of the system. The reliability aspect of the system is simulated by a model using Markov chains. Having defined the system state space and the transition rates, the model provides evaluation of the state probabilities. Evaluation of system utility is based on computer task values and the failure probabilities. Application of the analysis model to an existing system reveals information useful in assigning redundancy, eliminating bottle-necks and allocating spare parts.

INTRODUCTION

The trend towards still higher degrees of automation of machinery plant functions, has increasingly involved the electronic computer as an important active device onboard ships. Primarily, the computer is used to perform monitoring tasks as alarm and safety functions, but also to perform functions as condition monitoring of important components within the machinery plant, and active on-line tasks as bridge control functions, hull monitoring functions and loading/unloading calculations, to

mention a few.

Regarding the safety of a ship, one of the most interesting aspects of computerized ship functions is the supervision and automatic control of the machinery plant and especially of the propulsion machinery. In this respect, the requirements of Det norske Veritas as a ship classification society should be mentioned. Already in 1965, Det norske Veritas introduced as the first classification society, rules applying to the instrumentation of machinery plants, intended for periodically unattended operation. These rules are now extended to cover computer installations as well, and in this respect, reliability analysis has proved to be a useful tool.

The introduction of computers onboard ships poses new problems to be considered. The two most important problems are respectively integration of alarm and safety functions in the computer system and the complex environment which is encountered.
The latter problem mostly concerns installation techniques and environmental testing, however, the first problem is of a more philosophical nature regarding systems analysis and design. In this context it is felt that the reliability characteristics of computer hardware alone is not a satisfactory measure of system utility. It is therefore proposed that an approach where the computer system is considered as an integral part of the ship is more realistic when evaluating the utility of a computer system.

The proposed analysis method combines conventional reliability calculations and risk value evaluation into a utility simulation of the computer system, based on the operational characteristics. An important part of the analysis is the establishing of a model of the computer system.

METHOD OF ANALYSIS

The proposed analysis method is intended to be a simple and practical tool in evaluating the utility of shipborne computer systems. The main features are the simplicity of the analysis and the combination of economic and reliability characteristics to provide a better basis for decision making.
Fig. 1 shows some of the elements contained in the analysis.

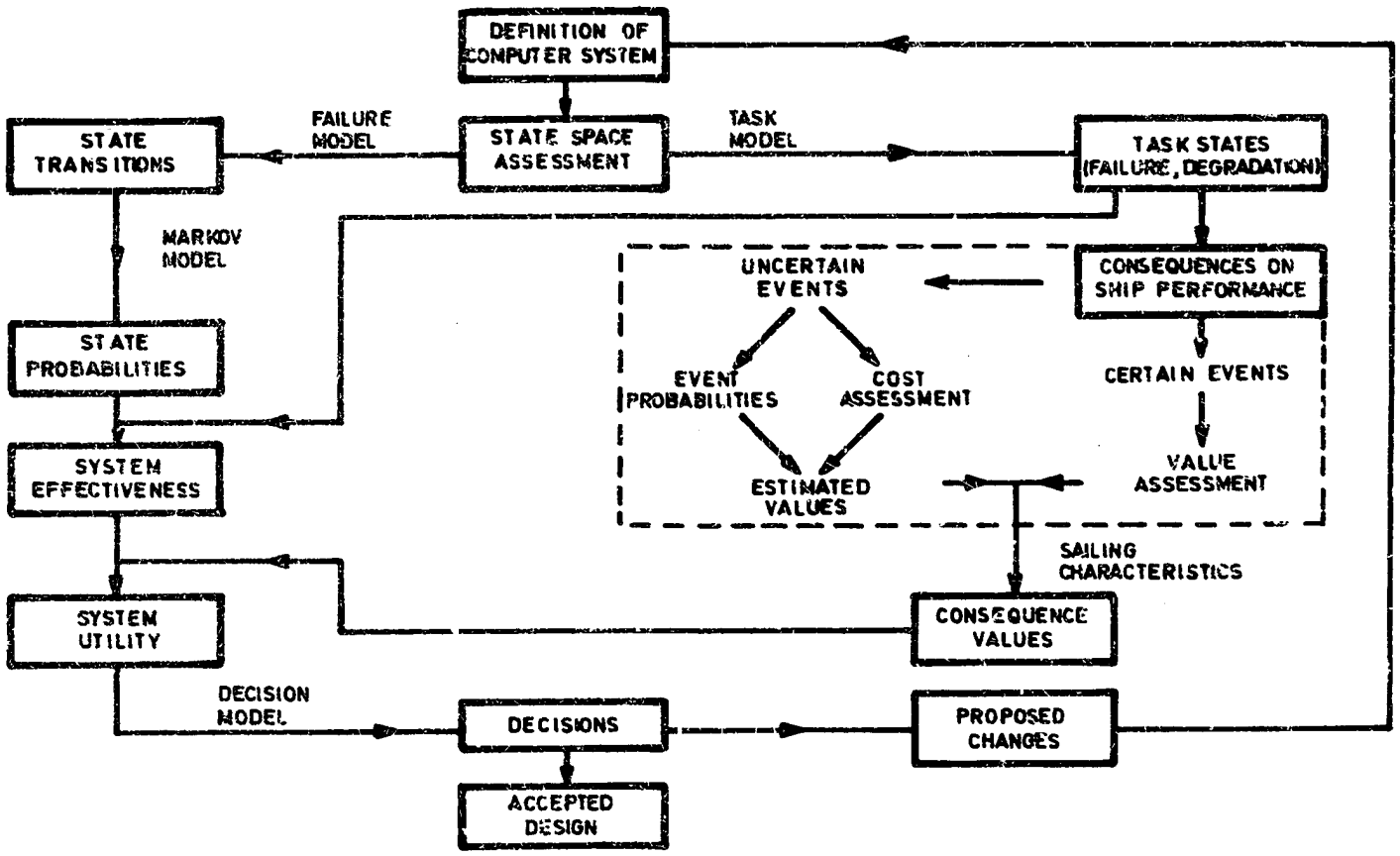


Fig. 1. Procedure for utility analysis of ship computer systems.

The probabilistic approach

Loss of ability to perform the intended tasks constitutes failure of the computer system. Failure of any unit or any combination thereof which can have this effect is considered to be a critical failure. However, failure in some units may only degrade system performance or reduce the instantaneous availability. The computer system thus enters different operational states, depending on the operational states of the units within the system or the transitions between these.

The dynamic behaviour of the system capability may be considered to constitute a stochastic process which actually is defined by the failure behaviour of its units. The operational states of the system are determined from the states of the hardware units and their software implementation. In this context, however, software failures are left alone, and the stochastic process only involves failure of the computer system hardware units. The computer software is looked upon as certain characteristics associated to unique or composite hardware states.

Defining a possible set of events implying system failure, it is possible to define a finite system space. The computer system is assumed to be composed of independent units, and each unit is considered to be in only two states; operating or failed. A system consisting of n independent units may therefore enter 2^n different states, and this state space describes the different fail-

ure or operational modes of the system. If the probability of being in a state only depends on the previous state and the transition probabilities between these states are independent of time, the behaviour of the system describes a Markov chain. Given the failure model of the intended unit, the Markov chain will simulate the dynamic behaviour of the system, i.e. the transitions between the possible system states and the probabilities of the different operational modes.

Computer task availabilities

The traditional definition of reliability is impractical to apply to a computer system. This definition says that reliability of some device is defined as the probability the device will perform its function without failure for a specified period of time under stated conditions. Because failure of hardware components does not automatically imply system failure or task failure, the concept of task availability is a better measure of system performance than hardware availability. Some of the system hardware states may leave a specified task completely intact, while others may only degrade the task performance. These states will be termed successful states for that particular task. The states completely destroying task performance will be term unsuccessful or critical states.

Let $P_j(t)$ be a column vector containing the probabilities of the successful states for task j at time t. Let D_j be another column vector containing numbers indicating the degree to which functional requirements of task j can be accomplished in state i, i $\in$ ⟨successful states⟩ Then

$$A_j(t) = P_j^T(t) \cdot D_j \qquad (1)$$

is a measure of the functional availability of computer task j at time t, given a specified software implementation. $A_j(t)$ may also be termed the system effectiveness for task j at time t. Give a mission time T, the system/mission effectiveness for task j is:

$$E_j = \frac{1}{T} \sum_{i=1}^{m} \left[\int_{t_i}^{t_{i+1}} A_{j,i}(t)\, dt \right]; \qquad (2)$$

where mission duration $T = \sum_{i=1}^{m} (t_{i+1} - t_i)$, m is the number of mission phases and t_i is the time at start of mission phase i.

Defining another column vector $\bar{D}_j = U - D_j$ where U is a unit vector, and substituting $\bar{D}_j$ for D_j in equation (1), gives:

$$\bar{A}_j(t) = P_j^T(t) \cdot \bar{D}_j : \qquad (3)$$

where $\bar{A}_j$ is a measure of the unavailability of task j at time t.

Tracing failure consequences

Extending the line of thought in evaluating task

availabilities to estimate total computer system performance availability by a nondimensional "percentage" value vector like vector D, does not give the information one needs for making decisions on system utility. In addition, the computer system application and operational environment have to be taken into account.

Given the computer system tasks, their degree of back-up and the operational profile of the ship, a value can be assigned to each task by analysing the consequences resulting from the loss of each task. The consequences concern the whole ship and take the shape of events which may be of a more or less disastrous kind. Some of these events are certain, and some are uncertain - the degree of uncertainty depending mainly on sailing characteristics, loading, geographical position and human interaction. The events can be damage to property, loss of time (off-hire) or inconvenience in the form of dispute, loss of reputation etc..

Tracing the possible sequence of events resulting from loss of a specified computer task, can be accomplished by means of a logical consequence diagram. An estimation of the event sequences, their duration and probabilities can be performed without any knowledge of the failure process of the computer system hardware. The consequence analysis is therefore preferably performed by people with an intimate knowledge of sailing ships and ship operation and with access to damage statistics. Usually several tasks can involve the same events, possibly with different probabilities. The different consequence diagrams thereby become coupled to each other.

Evaluation of system utility

All events resulting from computer task failure are supposed to have a value, either instantaneous or time dependent. These values are independent of how the task failures were initiated and they are estimated from the direct costs associated to an event. The more inaccurate status value of the circumstances connected to the event are also considered.

By collecting all events branching out from a task failure in a consequence diagram and making value estimates from event costs, probabilities and waiting times, a specific value is assigned to every computer task. Going back to the concept of task availability or system effectiveness, the task values are connected to the computer hardware as a measure of the failure consequences from hardware.

Defining the concept of utility as a numerical value of the prospect facing someone in a situation given certain assumptions, the task values can be interpreted as the utility of the different computer system hardware states, (1). Taking into account investment costs, the operating costs and the stochastic failure process of the

hardware units, enables the analyst to evaluate the total computer system utility. A utility appraisal of the system can also be done without regarding the investment and operating cost.

SIMULATION MODELS

An important part of the analysis is the simulation of the failure behaviour of the computer hardware system. This produces the system state probabilities which are used as input to the system effectiveness calculations.

The failure rate concept

If f(t) denotes the probability density function of a unit, then f(t) dt is the probability that the unit will fail in the time interval(t, t+dt). The probability that the unit will survive for the period (0, t) is then:

$$R(t) = 1 - \int_0^t f(x)dx = \int_t^{\infty} f(x)dx \quad ; \qquad (4)$$

which means that:

$$- \frac{dR(t)}{dt} = f(t) \quad ; \qquad (5)$$

The failure rate z(t) of the unit may be defined as the conditional probability that the unit will fail in a time interval (t, t+dt), given that it has survived up to time t:

$$z(t) = \frac{f(t)}{R(t)} = \frac{d}{dt} \left(\ln R(t)\right) \quad ; \qquad (6)$$

Assuming the hardware units in the computer system to be subject to chance failures only, the failure rate for each unit is assumed constant: $z(t) = \lambda$, and the expression of reliability in equation (4) becomes:

$$R(t) = e^{-\lambda t} \quad , \text{ since } R(0) = 1 \quad ; \qquad (7)$$

Because the conditional probability $z(t) \cdot dt = \lambda \cdot dt$ depends only on dt and is independent of t, the expected life time of a unit, MTTF, is constant at all times and equal to the reciprocal of the failure rate:

$$MTTF = 1/\lambda \quad ; \qquad (8)$$

This implies that if the independent units composing the computer system have exponentially distributed times to failure, then the time to system hardware failure will also be exponentially distributed. For repairable units, the assumption of constant failure rate λ and repair rate μ , means that operating time between hardware unit failures and the time required for repair (MTTR) of each unit composing the system, are exponentially distributed.

The Markov Model

Having defined a state space for the computer system, a Markov process is one whereby the system occupies a certain state and either undergoes a transition from this state to another, or remains in its present state with time homogeneous transition probabilities which only de-

pend on the previous state.

The Markov chain defined by a discrete state space and continuous time parameter is a stocastic model very suitable in describing the behaviour of complex systems.

Let $p_i(t)$ denote the probability that the system is in a state i at time t. For a state space containing a finite and countable number of states N, obviously

$$\sum_{i=1}^{N} p_i(t) = 1 \quad ; \qquad (9)$$

Let P(t) be a column vector whose elements are the system state probabilities at time t. P(t) may be called the state vector. The transition probabilities or rates in the Markov chains will consist of the repair and failure rates of the actual computer system as previously defined. The requirement of time homogenity is fulfilled by the exponential density functions for time to failure and time to repair. Use of the Chapman Kolmogorov differential equation gives

$$\frac{d}{dt} P(t) = (M) \cdot P(t) \quad : \qquad (10)$$

where (M) is the NxN matrix of the transition rates.

Knowing the initial conditions given by the state vector P(0), the set of simultaneous differential equations can be solved, and the probability vector for the system states is obtained as a function of time.

The transition rate matrix(M) is the basic element in the Markov model, and it characterizes both the system being analysed as well as the analysis.

If the computer system is repairable in all states containing failed hardware units, i.e. all states communicate, then the transition matrix and the states are called ergodic or positive recurrent. States which are not ergodic, are called transient.

In a completely ergodic process, the limits:

$$\lim_{t \to \infty} P_i(t) = P_i \qquad (11)$$

exist for all states i in the state space. As

$t \to \infty$, equation (10) becomes:

$$(M) \cdot P = 0 \qquad (12)$$

Together with equation (9) this equation implies that the limiting state probabilities can be determined by solving a set of linear algebraic equations.

A useful tool in Markov analysis is to prepare a diagrammatic representation of the transition rate matrix(M). The graph is called a reliability transition diagram, and it is composed of nodes representing system states and branches representing the possible transitions between the

states. Labelling the branches with transition rates makes it quite simple to evaluate the elements in (M). Examples of transition diagrams are given in fig. 4 and fig. 5.

Computer programs

In order to cope with the problems of solving the equation systems of equation (10) and (12 and 9) in a fast and economical manner, two computer programs have been developed, REAVAN and STAVAN (2).

The program REAVAN solves the set of differential equations given by equation (10), utilizing the Kutta-Merson algorithm. The result is the probability state vector P(t) as a function of time, for a finite time period with specified time intervals.

The program STAVAN solves the set of linear algebraic equations given by equations (9) and (12). The solution technique is based on an optimal Jordan elimination process, and the result is the steady-state probability vector P and the waiting times between different specified subsets of states.

Both programs are written in the ALGOL programming language for UNIVAC 1107 and 1108 computers with EXEC 8. Some of the subroutines involving manipulation of matrices are, however, written in FORTRAN IV. The programs have proved to be extremely helpful in evaluating system state probabilities. Computing time being only a few seconds, the programs are economical to run and give a lot of information in short time.

APPLICATION OF THE ANALYSIS METHOD TO A DESIGN STUDY

Given an actual ship and the tasks to be performed by the computer system, an example will be given, showing how use of the described method can be used to increase the utility of a system at the design stage. This is done by assigning redundancy, eliminate bottle-necks and allocate spare parts with respect to the ship's function and environmental conditions.

The ship system and cost values

The ship system to be considered is a machinery plant, with special emphasis placed on the propulsion machinery. Supervision and control of the machinery (referred to as the E0 tasks) and condition monitoring (referred to as CM) are the main tasks to be performed by the computer system.

The analysis method allows partition of the analysis into two groups, or submodels of the overall system. One consists of the computer system including the tasks to be performed. The other is the ship system which defines the computer tasks. Description of the ship system and the assigning of cost values to the different tasks to be performed by the computer system, will

not be shown here. The value estimation can best be done by personnel with experience in and knowledge of sailing ships and ship machinery plants, since the value estimation is independent of the computer configuration.

The cost estimation must take account of sailing schedules, harbourage, type of ship etc.. Stop of main propulsion involves greater risk, i.e. expected cost, to the ship when manoeuvring in restricted water than when sailing in open sea.

For the estimation of the different cost values, a typical voyage of 24 days in open sea, 4 days in restricted waters and two days in harbour is taken into consideration, (3).

The time dependent and immediate values for loss of computer tasks are shown in table 1. These values are valid for all four system alternatives outlined in the following.

The basic computer system

The starting point in this design study is the basic computer system A, shown in fig. 2. It consists of a computer (COM) (CPU, memory, interface for typewriter, punch, tapereader, computer operator panel etc.) and a typewriter (TW). Further there is a control console (CC) connected to the computer through the process input/output system (PIO). A display (CRT) is also connected through the PIO. A tapereader (TR) is used for loading programs into the computer, and the whole computer system is fed by power from the main switchboard (MSB). No tapepunch is shown since it is not necessary for the overall system function.

The failure and repair data used in this analysis are estimated after communication with designers of related systems. The main input data to the computer programs REAVAN and STAVAN are the mean time between failure (MTBF) and the mean time to repair (MTTR) for each component in the system. In table 2, the columns 3 and 4 show data valid for the basic system A. (The table also includes data used for the systems B, C and D.)

The basic system is supposed to consist of seven independent units (see fig. 2). Each unit is considered to be in only one of two states, operating or failed. The system may therefore enter $2^7 = 128$ different states. Since the MTBF is much larger than the MTTR for all units in the system, every combination of unit failures yielding consequences less severe than the consequences of each subset within the combination are neglected. Figure 4 shows the reliability transition diagram for the basic system A. Only 9 states are considered to be of interest. The states 2, 3, 4 and 5 will cause loss of all computer tasks.

Some results obtained from the computer program REAVAN are shown in fig. 6, 7 and 8 for all four system configurations. In fig. 6a, the

dynamic behaviour of the probabilities of system success are plotted. The steady state availabilities for repairable systems are reached in approximately 8 to 12 hours after starting the systems. Fig. 6b shows the corresponding probabilities of computer failure which is a critical failure mode. In measuring the utility value of the system, a.o. equation (3) is used to compute the task value function for the system. The simple decision table, table 3, shows the connection between hardware failures and total or partial loss of computer tasks. Decision tables are used to prepare information for input to computer programs calculating utility values.

The calculated utility values for loss of system performance are shown in fig. 7. In fig. 8, the task availabilities are shown as calculated by the computer program STAVAN.

Analysis method

The three system configurations B, C and D are modifications of basic computer system A. The objective is to improve the availability of the computer tasks, thus decreasing the overall risk utility. Experience has shown that the power supply from the main switchboard is critical. Use of this power supply causes the MTTF for several units, especially the computer and tape-reader, to decrease to a value much below the corresponding value for land-based computer systems.

Feeding power continuously through a battery bank to the computer system, improves the MTTF for several units, see 5th and 6th column in table 2. Additionally, the battery power supply guarantees the system continuous power for at least 30 min. if a main switchboard break-down occurs. The addition of a battery supply to system A gives system B.

Table 4 shows that the E0 and CM tasks depend heavily on precise function of the typewriter. The table also shows that no reconfiguration of the program system can be performed without proper function of the tape-reader. Usually, typewriters are equipped with slow tape-readers. Modifying the software and hardware system in such a way that the tape-reader on the typewriter can be used as back-up, and adding an extra typewriter for redundancy, we call the new configuration system C.

Again, calculations on the modified computer system show an improvement of the utility function in spite of a small decrease in the steady state and dynamic availability of the computer hardware. According to table 5, the only "bottle-necks" remaining are the computer itself and the process input/output system. These are the only units which by a single failure can cause total system break-down.

The failure consequences presented in table 1, show that the E0-tasks are much more impor-

tant than the CM-tasks. Using two computers, one for the E0-tasks and the other for the CM-tasks, are giving back-up for the E0-computer tasks at the expense of the CM-tasks. This also results in a higher MTTF for each computer in this new system compared to the computers in system A, B and C, owing to reduction of the memory capacity of each computer. Using this modification, a data channel (ACM) is needed for communication between the two computers.

In the process I/O system, the multiplexers and converters are some of the most unreliable parts. The I/O system is divided into three parts. Two identical parts, containing the most unreliable part of the I/O system, serve each of the two computers. The part serving the less important computer will serve as a stand-by unit for the most important computer (E0-tasks). The third part of the I/O system is quite reliable, so this remaining bottle-neck is acceptable from a reliability point of view.

A block diagram for this system, containing 13 units, is shown in fig. 3. Theoretically, the system can enter $2^{13} = 8192$ different states, but without loss of any significant information, the method applied allows for a reduction to only 28 states. The reliability transition diagram for system D is shown in fig. 5.

The improvement in risk utility from system A to system D is shown to be a factor of 2.5 (fig. 7), with a corresponding increase in task availability.

Hitherto, we have assumed that all system failures have been repairable with a MTTR given in table 2. This will not always be true, especially for computer systems onboard ships due to lack of specialists, tools, spare parts etc.

In order to demonstrate a way of allocating spare parts, calculations have been performed assuming that the CRT display, battery power supply and two identical parts of the I/O system are not repairable (absorbing states). The results are plotted into fig. 6a.

CONCLUSION

In the preceeding sections, an analysis method intended for evaluation of computer systems on board, has been presented. The procedure may seem somewhat complex at first, but it has been found to be a simple and efficient way of obtaining information on computer system structures. It is felt that reliability data alone are not satisfactory as a basis for selecting between alternative system configurations. The concepts of utility and task values, however, prove to provide information relevant to systems evaluation and design.

Application of Markov models has been found to constitute a very convenient analysis tool in

systems design, because:

- The concept is easy to understand.
- The model is easy to use.
- The state space is easy to change.
- The system structure is easy to change.
- Alternative systems are easily compared.
- Sensitivity analysis is easy to perform.
- Computer analysis takes only a few seconds.

Even if the Markov chains in some cases may not be the correct stochastic description of the system, it still gives information enabling comparative analysis of systems.

	E0	CM
Immediate value	$2.7 \cdot 10^4$	0
Time dependent value	$97 \cdot 10^4$	$0.21 \cdot 10^4$

Table 1. Value for computer tasks.

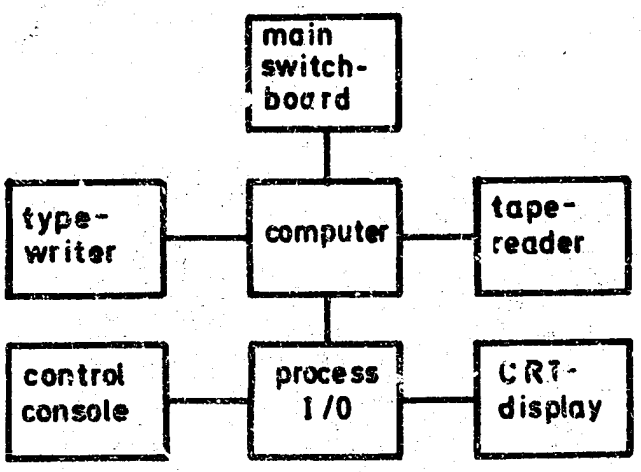


Fig 2. Basic computer system A

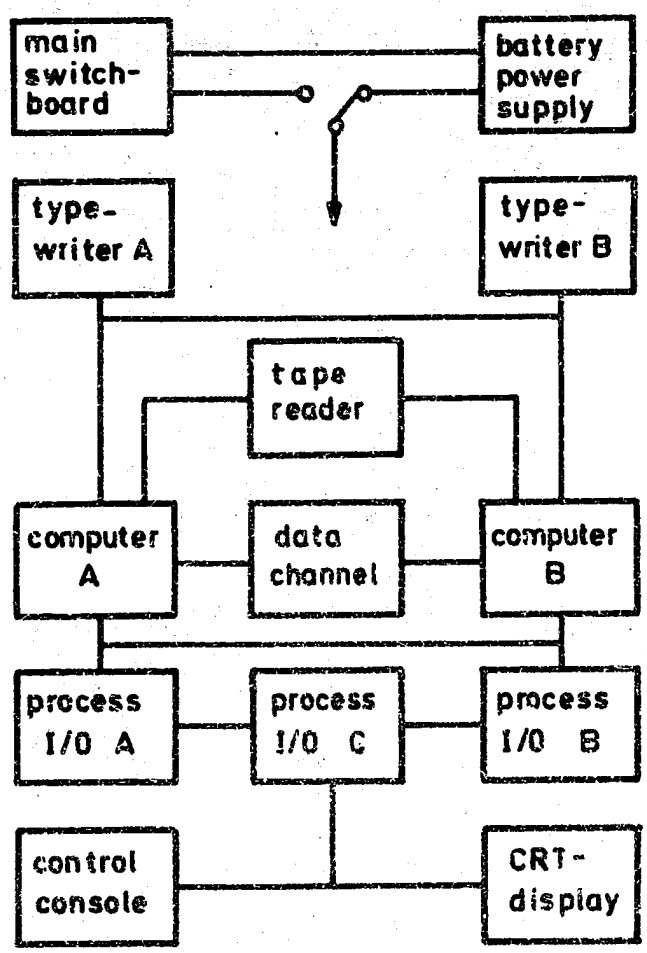


Fig. 3. Computer system D

REFERENCES:

(1) H. Chernoff and L. E. Moses: "Elementary Decision Theory". John Wiley & Sons Inc., New York 1959.

(2) Tor-Christian Mathiesen: "Reliability Engineering and Ship Machinery Plant Design". Lic. Techn. Thesis.

(3) T. Heimly, G. Dahll, C. Bøe: "Reliability and Availability of Computer Systems onboard Ships" (In Norwegian). Report from Det norske Veritas, Machinery department 1972.

	Component		using Main switchboard MTTF (hours)	using Main switchboard MTTR (hours)	using Battery power supply MTTF (hours)	using Battery power supply MTTR (hours)
	Computer (system A, B and C)	COM	1250	4	2000	4
	Typewriter	TW	1000	2	1000	2
	Tapereader	TR	1/2*)	5	1/10*)	5
	Control console	CC	25000	10	30000	10
	Process I/O	PIO	8000	6	10000	6
	CRT-display	CRT	2000	8	3000	8
	Main switchboard	MSB	1000	1	1000	1
	Battery power supply	BPS	1000	3	1000	3
	Computer using tapereader on typewriter			8		10
System D	Data channel	ACM	10000	5	20000	5
System D	Computer A	COMA	2500	4	5000	4
System D	Computer B	COMB	2500	4	5000	4
System D	Process I/O, part A	PIOA	8000	6	10000	6
System D	Process I/O, part B	PIOB	8000	6	10000	6
System D	Process I/O, part C	PIOC	40000	6	40000	6

*) failure every 2nd or 10th time when used.

Table 2 Failure and repair data

COM	y						y				ELSE
PIO		y									
MSB			y							y	
TW				y				y			
CC					y						
CRT						y		y			
TR							y				
BPS									y	y	
EO	1	1	0	.2	.4	0	1	.2	.1	1	E
CM	1	1	0	.6	.4	.5	1	.8	.1	1	E

Table 4. Decision table, system B (E = error, considering not allowed states)

COM	y											y		ELSE
PIO		y												
MSB			y										y	
TWA				y							y			
TWB					y						y			
TWA or TWB						y								
CC							y							
CRT								y						
TR									y			y		
BPS										y			y	
EO	1	1	0	0	0	0	.4	0	0	.1	.2	.1	1	E
CM	1	1	0	.4	.4	.4	4	.5	0	.1	.6	.1	1	E

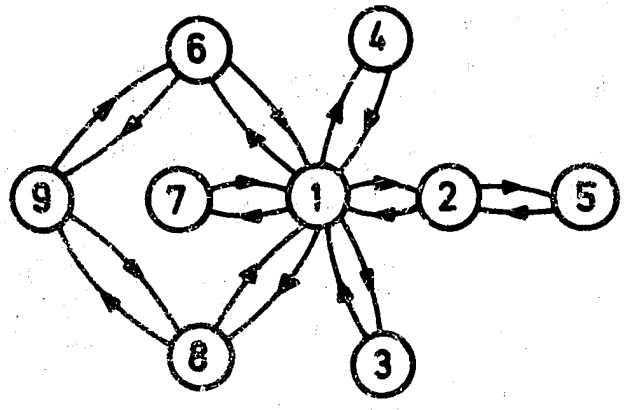
Table 5. Decision table, system C

COMA	y															y			y		ELSE
COMB		y														y					
PIOA			y														y				
PIOB				y													y				
PIOC					y																
PIOA or PIOB						y															
MSB							y													y	
TWA								y										y			
TWB									y									y			
TWA or TWB										y											
CC											y										
CRT												y									
TR													y						y		
ACM														y							
BPS															y					y	
EO	0	0	0	0	1	0	0	0	0	0	.4	0	0	0	.1	1	1	.2	.1	1	E
CM	1	1	1	1	1	1	0	.4	.4	.4	.4	.5	0	1	.1	1	1	.6	.1	1	E

Table 6. Decision table, system D

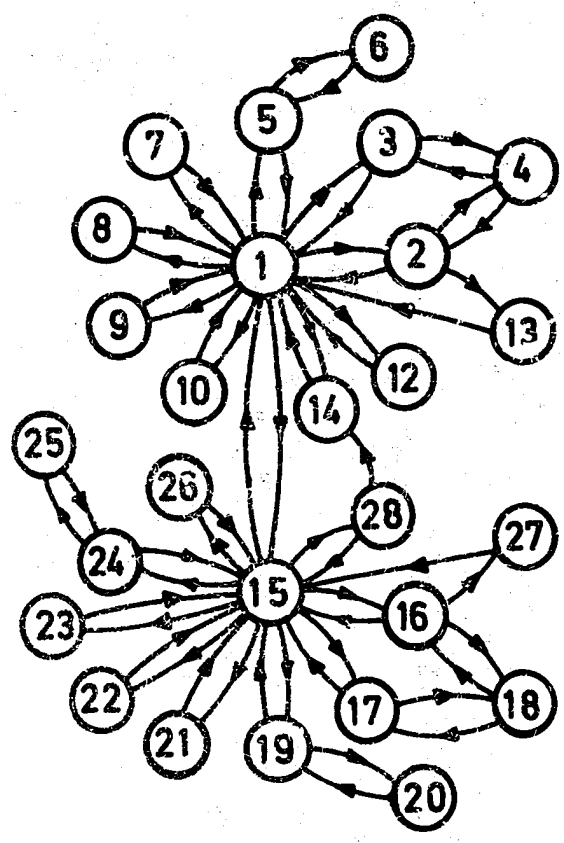
COM	y							y		ELSE
PIO		y								
MSB			y							
TW				y					y	
CC					y					
CRT						y			y	
TR							y	y		
EO	1	1	1	.2	.4	0	0	1	.2	E
CM	1	1	1	.6	.4	.5	0	1	.8	E

Table 3. Decision table, system A



1. System o.k.
2. Computer fault
3. Process I/O fault
4. Main switchboard fault
5. Computer and tapereader fault
6. Typewriter fault
7. Control console fault
8. CRT-display fault
9. CRT and typewriter fault

Fig. 4. Reliability transition diagram for system A.



1. System o.k.
2. Computer A fault
3. Computer B fault
4. Computer A and B fault
5. Typewriter A or B fault
6. Typewriter A and B fault
7. Control console fault
8. CRT-display fault
9. Process I/O C fault
10. Process I/O A or B fault
11. Process I/O A and B fault
12. Data channel fault
13. Computer A and tapereader fault
14. Main switchboard fault
15. Battery power supply fault

Battery power supply fault:

16. Computer A fault
17. Computer B fault
18. Computer A and B fault
19. Typewriter A or B fault
20. Typewriter A and B fault
21. Control console fault
22. CRT-display fault
23. Process I/O C fault
24. Process I/O A or B fault
25. Process I/O A and B fault
26. Data channel fault
27. Computer A and tapereader fault
28. Main switchboard and battery power supply fault

Fig. 5. Reliability transition diagram for system D.

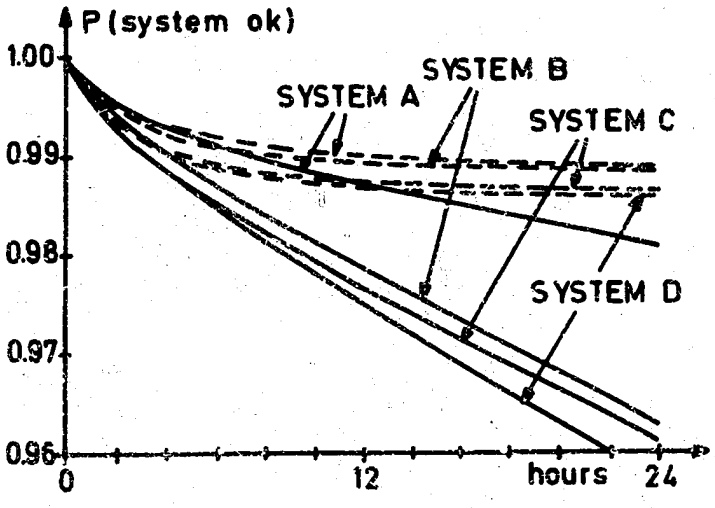


Fig. 6a Dynamic behavior of systems as calculated by REAVAN

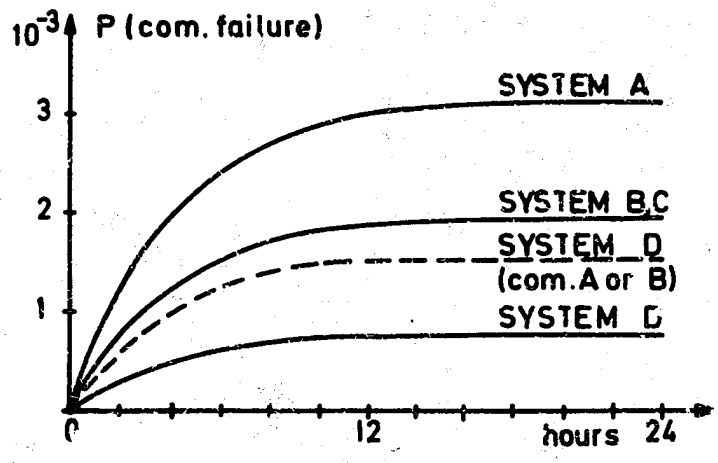


Fig. 6b Probability of computer failure as calculated by REAVAN

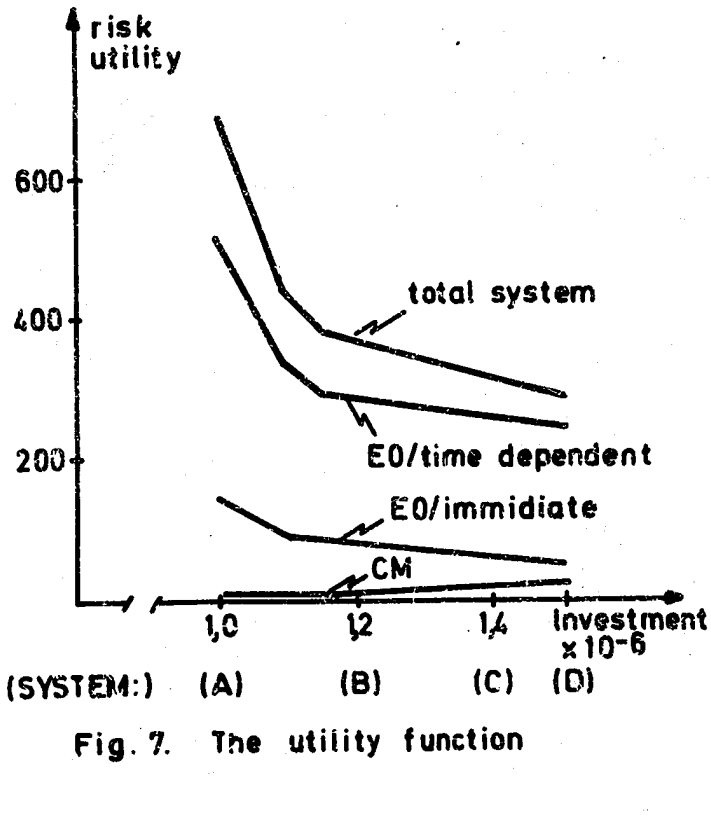


Fig. 7. The utility function

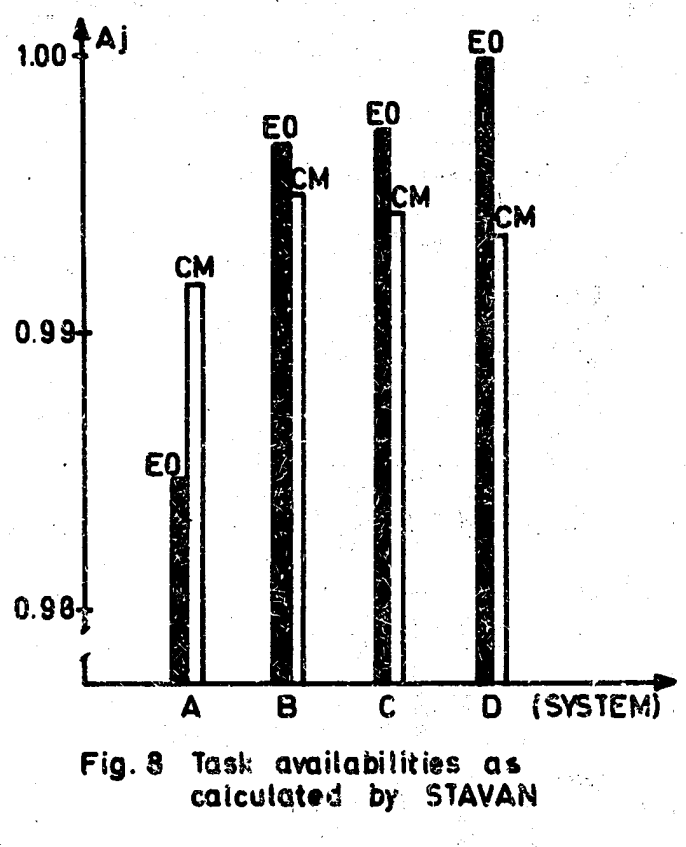


Fig. 8 Task availabilities as calculated by STAVAN