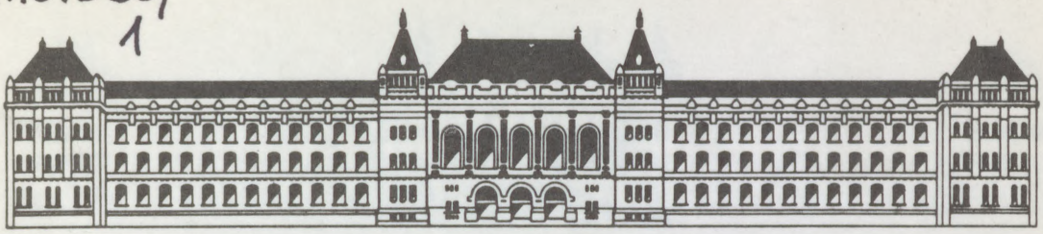


TECHNICAL UNIVERSITY OF BUDAPEST

P R O C E E D I N G S

of the

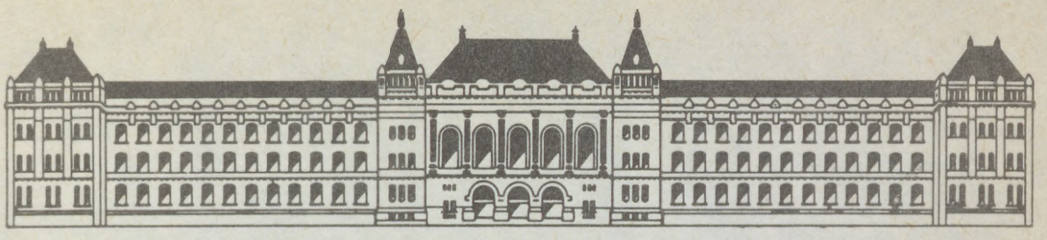
ADVANCED STUDIES ON RELIABILITY ENGINEERING (A.S.R.E.)

Summer course at the Technical University of Budapest

3-7 September, 1990.

Organizers: the Institute for Precision Mechanics and Optics
the Institute for Communications Electronics
the International Center for Engineering Programs

Sponsor: Digital Equipment (Magyarországi) Kft.



TECHNICAL UNIVERSITY OF BUDAPEST

P R O C E E D I N G S

of the

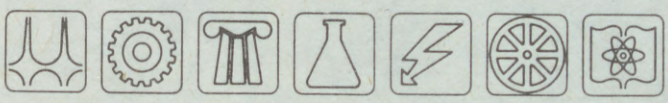
ADVANCED STUDIES ON RELIABILITY ENGINEERING (A.S.R.E.)

Summer course at the Technical University of Budapest

3-7 September, 1990.

Organizers: the Institute for Precision Mechanics and Optics
the Institute for Communications Electronics
the International Center for Engineering Programs

Sponsor: Digital Equipment (Magyarországi) Kft.



P R O C E E D I N G S

of the

ADVANCED STUDIES ON RELIABILITY ENGINEERING (A.S.R.E.)

Summer course at the Technical University of Budapest

3-7 September, 1990.

ISBN. 963.420.234.9

ISBN. 963.420.237.3

Organizers: the Institute for Precision Mechanics and Optics
the Institute for Communications Electronics
the International Center for Engineering Programs

Sponsor: Digital Equipment (Magyarországi) Kft.

P R O C E E D I N G S

of the

ADVANCED STUDIES ON RELIABILITY ENGINEERING (A.S.R.E.)
Summer course at the Technical University of Budapest

CONTENTS

Invited Papers

Abstracts of contributed papers

The Organizing Committe of the course expresses its regrets for all papers and abstracts which could not be included in this volume due to missing the deadline, 31th July, 1990.

INVITED PAPERS

DESIGN ASPECTS OF RELIABILITY PRODUCTION OF SIMULATION EQUIPMENT

Alan FOWLER, Link-Miles Limited

United Kingdom

ADVANCED STUDIES ON RELIABILITY ENGINEERING

at the

TECHNICAL UNIVERSITY OF BUDAPEST

3 - 7 September 1990

Summer Workshop

Subject of Lecture

Design aspects of Reliability Production of Simulation Equipment.

Presented by

Alan Fowler C.Eng., M.I.Prod.E., F.I.Q.A.
Product Assurance Manager
Link-Miles Limited
Lancing, West Sussex, England.

Contents

1.0 INTRODUCTION

Today's markets are extremely discriminating and customers demand product performance that is equal to or better than the declared specification, or in the case of the open market, a product performance that is equal to or better than the consumer's expectations.

In order to meet these requirements, considerable thought and effort has to be applied by the Project and Design teams involved in order to plan how they are going to achieve the expected performance criteria.

In all cases 'Reliability' is a key parameter that has to be addressed and this paper outlines some of the techniques and methodologies available.

For the purposes of this paper the definition of Reliability is as follows:

Reliability

The ability of the product or service to perform to specification under stated conditions for a stated period of time.

2.0 RELIABILITY PROGRAM

2.1 General

Reliability is a principal Quality Measure and the purpose of a Reliability Program is to identify those aspects of reliability that require to be addressed through the life cycle of the product from customer definition stage through design, development, manufacture and test to Customer Acceptance.

A properly integrated reliability program will provide an overall cost benefit to the Company concerned in terms of reduced costs in design rework and reduced maintenance costs when in service.

2.2 Selection of program activities

It is not possible to say in precise terms the effectiveness of each program activity on overall product reliability.

The choice of activities and resources to be expended should be based on previous experience. If no previous experience exists then the choice of activities undertaken should be made on a "common sense" approach as follows (BS5760 refers):

a) The earlier a design change can be agreed, the lower the total cost is likely to be. The cost of making a design change at the production or use stage is usually many times the cost of doing so at the initial design and development stages.

b) For complex items, design evaluation and review, use of proven parts and processes, and the use of redundancy or derating within the design can greatly enhance overall system reliability. Large scale use of system redundancy can greatly improve overall system reliability, but at an increase in initial cost and maintenance load.

c) Reliability testing, including corrective action, carried out during the development and production phases, is an effective means of increasing the reliability of complex items as well as increasing confidence that the required level has been attained.

d) Reliability improvement depends on several factors such as the effectiveness of procedures for identifying, reporting and taking action on failures, the way the programme is managed and the use to which failure analyses are put.

2.3 Typical Reliability Program

See figure (1).

3.0 RELIABILITY SPECIFICATION

3.1 Introduction

There are many ways in which a customer can express reliability requirements and these are normally linked to operational constraints such as the Duty Cycle, Maintenance Policy and Project Life.

3.2 Typical Parameters

a) Failure Rate (FPMH)
The failure rate is usually in units of failures per 10^6 hours and is the summation of the individual failure rates comprising the product.

b) Mean Time Between Failure (MTBF)

$$MTBF = \frac{1}{FPMH} \text{ (hours)}$$

c) Reliability (R)

$$R = e^{-\lambda t}$$

where λ = Failure Rate
t = Mission Time

d) Mean Time to Repair (MTTR)
This is simply the repair time that the customer expects not to be exceeded to return the product to full working condition. The MTTR calculation includes various factors including:
- Diagnostics
- Removal
- Repair
- Refit
- Test, etc.

In some circumstances there are four MTTR's quoted and these are known as first, second, third and fourth line repairs which relate to where the repairs are carried out, the degree of work to be conducted and the skill levels required.

e) Availability (A)

$$A = \frac{MTBF}{MTBF + MTTR}$$

where MTBF = mean time between failure, and
MTTR = mean time to repair

f) Life (L)
The expected time that the product is expected to remain in service prior to replacement or major overhaul.

g) Mission Time (t)
The duration in hours over which the product is expected to achieve the Reliability.

h) Duty Cycle or Operational Profile
A statement is required from the customer, defining the 'Duty Cycle' or 'Operational Profile' of the product, as this is a primary reliability consideration.

For Example:

A typical duty cycle for an office computer may be:

0 to 0800 hours	-	Switched off
0800 to 1200 hours	-	Full Operation
1200 to 1300 hours	-	Stand by
1300 to 1800 hours	-	Full Operation
1800 to 2400 hours	-	Switched off

3.3 Worked Example

A system has an MTBF of 200 hours and is required to have a Reliability of 97% over a 5 hour period and an availability of not less than 98%. Calculate -

a) The 'failure rate'
b) The 'reliability'
c) The 'availability' where the time to repair is 2 hours

a) Failure Rate $= \frac{1}{MTBF} = \frac{1}{200} = 0.005$

b) Reliability $= e^{-\lambda t} = e^{-0.005 \times 5} = 0.975$

$= 97.5\%$

c) Availability $= \frac{MTBF}{MTBF + MTTR} = \frac{200}{200 + 2} = 0.99$

$= 99\%$

4.0 RELIABILITY PLAN

4.1 Introduction

Reliability Plans define the tasks to be implemented to ensure that the specified requirements are met. The Plan is also used to monitor progress throughout the design and development phases and the tasks defined form an integral part of the overall engineering process.

In practice Reliability Plans cover Reliability and Maintainability aspects.

4.2 Contents

The contents of a Reliability Plan is usually as follows:

1. Introduction
2. Applicable Documents
3. Management Structure and Interfaces
4. Reliability and Maintainability Program

The Reliability and Maintainability Program covers the following elements.

4.3 Reliability Risk Areas

Reliability risk areas will be identified at outset of the project and monitored throughout the programme to ensure that visibility of these potential problems remains high. Any additional problems identified during the programme will be highlighted in a timely manner in order to minimise any impact on the reliability and maintainability goals set for the programme.

4.4 Project Progress Report

The Reliability Engineer will prepare progress reports for inclusion in the Project Progress Report. This report will cover the activities during the reporting period and outline the activities for the next reporting period.

4.5 Reliability Programme Review

The purpose of this review is to discuss all pertinent aspects of the reliability programme in order to ensure that it is proceeding in accordance with the contractual requirements and that all reliability requirements are being attained. This internal management review is held on a regular basis and is attended by project quality and reliability personnel.

4.6 Design Review

Design reviews are held regularly as part of the design process. A representative of the Reliability Group will attend all such design reviews to ensure that reliability is considered in the broad context of the overall design.

Check lists are used to assist in assessing the design and cover such topics as design simplification, component standardisation, derating, correct component application, component protection, minimum operator error and testability.

4.7 Parts Programme

All components to be used are selected in accordance with a comprehensive list of preferred parts and materials.

4.8 Reliability Model and Prediction

In order to assess the extent to which design criteria and reliability requirements have been incorporated into the design a reliability prediction will be generated.

This prediction will be based on a simple mathematical model derived from the breakdown of the defined system. The prediction technique requires that for each part and assembly a reliability figure is determined. Each part and assembly is analysed to determine if it is similar to any existing hardware with established reliability. If it is similar, the part of assembly is given the corresponding failure rate. If it is not similar, the part or assembly failure rate is estimated based on the techniques of MIL HDBK 217E.

The prediction will be refined as the project progresses and be presented in the form of a report to the customer prior to final acceptance.

The Stress Ratio criteria employed are summarised below:

Part Category	Stress Ratio	Base
Integrated Circuits	80%	Power Dissipation (maximum rated)
Transistors	50%	Power
	75%	Voltage
	75%	Current
Diodes and Rectifiers	50%	Power
	50%	Current
	75%	Voltage
Resistors	50%	Power
	80%	Voltage
Capacitors (except tantalum)	50%	Voltage
Capacitors (tantalum)	80%	Voltage
Switches	40%	Current
Relays	40%	Current

4.9 Failure Reporting, Analysis and Corrective Action System

During each phase of the manufacture, commissioning and acceptance testing of the system, procedures will be set up to collect, analyse and correct failures. This data covers both hardware and software. This data will be used to monitor the achievement of reliability during development.

4.10 Reliability Growth Measurement

Reliability growth is the improvement of equipment reliability over a period of time effected by systematic and permanent removal of failure mechanisms. The techniques to measure reliability growth are well established and documented, e.g. MIL HDBK 189. At the end of each acceptance phase the reliability growth data or simple estimates will be offered as demonstrating progress towards the achievement of the reliability specification. Simple estimates may be presented if there is limited data or if the reliability growth models are unable to give a reasonably goodness of fit measure.

During in-house testing reliability data will be collected and failures will be categorised into relevant and non-relevant for the purposes of reliability growth plotting. Also the operating hours will be recorded to enable plots to be completed of failures against time and failure rate against time. Reliability growth models will be used to measure and predict the reliability achieved.

4.11 Software Reliability

Software is one of the main contributors to system performance and is critical to the overall system reliability. Software failure depends on the quantity of inherent design faults (bugs) and the probability that they will affect performance. If these bugs are corrected properly, the software reliability is improved and can theoretically achieve 100%. However, to design, develop and test the software a Software Control Plan will be written detailing the modular approach to the structure and testing of the software. The Plan will also outline the controls and documentation that will be used to identify and overcome software problems.

4.12 Availability, Reliability and Maintainability Demonstration

The purpose of the AR&M demonstration is to show that the AR&M parameters in the specification have been achieved. An AR&M Demonstration Plan will be written and agreed with the Authority. This Plan will outline the documentation, the procedures and the conduct of the demonstration.

5.0 DESIGN FOR RELIABILITY

5.1 Introduction

The achievement of product reliability does not occur by accident, and the designer must not only recognise the rules associated with design philosophy but also make provisions for completing the design work, at the earliest possible stage in the program.

With many programs the major decisions such as outline performance, size, weight, major component procurements etc. are in fact made prior to contract award and the effect of these decisions often mean that they cannot be reversed. This again emphasises the need for those engineers responsible for Reliability to be party to the decisions made.

5.2 Design Philosophy

The design philosophy outlined below should apply to all products but clearly some degree of 'tailoring' will be needed to suit the specific needs of different products.

To achieve reliable products the designer should:

a) Understand the Product Reliability Specification
b) Recognise the environment in which the product is required to work
c) Use suitable components.
d) Keep the design simple.
e) Keep innovation to a minimum and maximise the use of proven technology.
f) Ensure that component stresses are suitable.
g) Design for ease of production, test and maintenance.

6.0 DESIGN REVIEW

6.1 Introduction

With the Customer's requirements clearly identified and Project Planning complete, the various elements of the product requiring design should in turn be identified.

The elements of the product requiring to be designed need to be 'broken down' into manageable tasks each with a responsible engineer, a specification, start/end dates to completion, and a means of measuring the success of the various tasks.

The responsibility for achieving 'quality of design' of course rests with the design team, whilst the Team Leaders would have the specific task of monitoring the above process. In addition it would be normal to set up an independent review body to provide assurance that the design objectives are being or have been met.

This independent activity is known as Design Review

6.2 Objective

Design Reviews are operated to a defined company procedure and call for the Design Team to present their work to a body of experts who are knowledgeable on the produce and would as such normally have a relevant field of expertise.

The Design Review Team would be expected to confirm or otherwise that the design objectives have been, or are in the process of being met and that any potential risk areas are identified together with any corrective action plans.

6.3 Factors considered

The factors considered by the Design Review Team would include:

a) The use of Company "Design Codes of Practice"
b) The use of "preferred" parts
c) The use of existing design solutions, thus minimising innovation and the use of untried technologies.
d) Identification of potential problem areas and/or uncertainties
e) A record of design decisions
f) Objective evidence

6.4 Structure

Design Reviews would normally be structured so that there is formal review at the beginning of the design process i.e. Preliminary Design Review, in the middle i.e. Critical Design Review and at the end i.e. Final Design Review. In between these reviews it would be normal for a schedule of lower level reviews to be held by the design team. The independent experts previously mentioned would normally only be involved in the three major reviews.

As general guidance, the three major reviews would endeavour to cover the following points:

Preliminary Design Review	Specification Delivery Schedule Design Philosophy Problem Areas Quality Program
Critical Design Review	As above Design Concept Technical Viability Verification and Validation Interfaces Compliance Check List
Final Design Review	As above Design Freeze Product Support Handbooks/Data Package

6.5 Design Team

A last word about the management of design teams. On one hand we have the structured approach to take into account that it is necessary to transfer the solutions in the designers mind to something that is a workable entity drawings or software, and on the other it has to be recognised that to achieve the best results of some designers and some situations, there may have to be a reasonable degree of freedom given that is later followed up by the more structured approach, but this essentially becomes a management issue.

However, it remains of vital importance that time is taken out to explain to the whole of the Design Team that which is expected of them as well as the various disciplines which apply.

7.0 SOFTWARE RELIABILITY

7.1 Introduction

Software is a common part of most systems today either the software is an integral part of the product or it is used in a supporting role. The question that is most commonly asked is 'How Software can be addressed in a Reliability Program?'.

Before this can be answered we first have to look at the various Reliability Terms and then how they apply to Hardware and Software.

7.2 Reliability Terms

	Hardware	Software
a)	**Failures** may be due to specification errors in design, production use, and maintenance.	**Failures** are due to design errors.

b)	Failures can be due to **wear out**, or other energy-related process. Sometimes warning is available before failure occurs as a result of gradual degradation.	There is no **wear out** process in software. Failures occur without warning.
c)	**Repairs** can be made which return the equipment to the original working state and may make the equipment more reliable.	**Repairs** cannot usually be made as they can for hardware. The main solution is usually to re-design, i.e. re-program, which, if it removes the error and introduces no other, will result in higher reliability in the eyes of the user. However, where data has been corrupted and 'repaired' the cause of the corruption must be sought and removed in order to prevent a reoccurrence.
d)	**Reliability** can depend upon 'burn-in' or wear-out processes, i.e. failure rates can be decreasing, constant, or increasing with respect to operating time.	**Reliability** is not so dependent. Reliability improvement over time may be affected, but this is not an operational time relation. Rather it is a function of the effort put in to detecting and correcting errors.
e)	Reliability is **time-related** with failures occurring as a function of operating and storage time.	Reliability is not calendar **time-related.** Failures occur when a program step or path which is in error is executed.
f)	Reliability is related to physical **environmental** factors such as temperature, humidity, and pressure.	Reliability is not affected by the physical **environment**, unless program inputs are affected.
g)	Reliability can be **predicted** in theory from knowledge of design and extent of use.	Reliability cannot be **predicted** from any physical bases, since it depends entirely on human factors in design.
h)	Reliability can sometimes be improved by redundancy.	Reliability cannot be improved by redundancy if parallel program paths are identical, since if one path fails any other will have the same error. It may be possible to provide redundancy by having parallel paths each with different programs written and checked by different teams.

7.3 Summary

We can see from the above that the traditional means of dealing with Hardware Reliability does not entirely apply to Software.

However, this does not mean to say that Software Reliability cannot be influenced in the Reliability Program Plan.

The measures that are recommended to ensure Software Reliability are as follows:

a) Define the Specification.
b) Define resources needed.
c) Plan the software life cycle i.e. design, development, testing etc.
d) Use Design Codes of Practice
e) Use previously 'tried and tested' software where possible.
f) Employ configuration control.
g) Correct problems in a controlled manner.
h) Test against the agreed specification.

When the product is in the hands of the customer it is likely that problems will manifest themselves and to enable realistic system level reliability predictions to be carried out then a failure rate allocation must be made for the software elements.

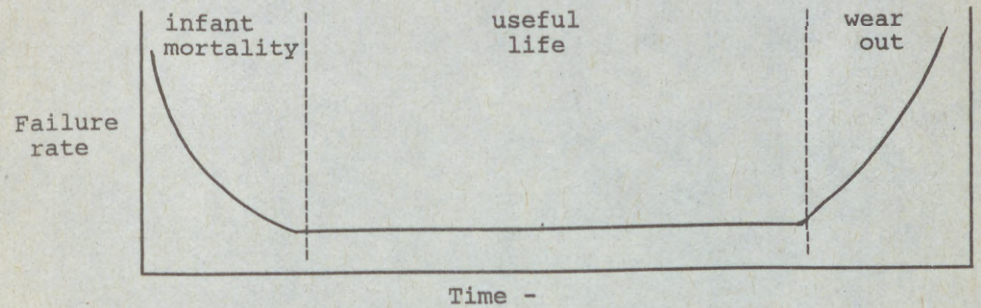
8.0 RELIABILITY ANALYSES

8.1 Introduction

There are various forms of reliability analyses that can be applied to a product or system and this section of the paper deals with some of the methods commonly in use.

8.2 The 'Bath Tub' Curve

The 'Bath Tub' Curve characterises any approximate reliability life cycle for mechanical, electro mechanical and electronic products and is made up of three distinct regions.



a) The 'Infant Mortality' region
This is caused by the presence of failures as a result of poor design or manufacture and is represented by a decreasing failure with respect to time.

b) The 'Useful Life' region
This part of the curve represents the period of the product life where the failure rate is constant and where predominantly random failures occur.

c) The 'Wear Out' region
The last part of the curve is known as the wear out region and this is characterised by an increasing failure rate with time.

8.3 Reliability Predictions

Reliability Predictions are used at the earliest possible stage of the design cycle in order to determine whether the equipment Mean-time-between-failure (MTBF) will be achieved.

Predictions will also identify potential weak areas of the design thus enabling timely corrective action to be taken.

To produce a reliability prediction the following information is needed:

a) Part failure rates, which are usually quoted in failures per million hours (FPMH) and are denoted by the Greek Letter LAMBDA (λ).

b) Environmental consideration i.e. temperature, stress, environmental factor.

c) Duty cycle, i.e. time in use during the day.

d) Equipment schematic showing the interrelationship of the various pieces of the system.

It is important to understand the source of the failure rate data in order that accurate comparisons of equipment can be made.

An example of a Reliability prediction is as follows:

$$\text{MTBF} = \frac{1}{\text{Sum of Failure Rates}}$$

$$\lambda 1 = 5 \times 10^{-6}$$

$$\lambda 2 = 2 \times 10^{-6}$$

$$\lambda 3 = 6 \times 10^{-6}$$

$$\lambda 4 = 21 \times 10^{-6}$$

$$\text{MTBF} = \frac{10^6}{5 + 2 + 6 + 21}$$

$$\underline{\text{MTBF} = 29412 \text{ hours}}$$

8.4 Failure Mode Effect, and Criticality Analysis (FMECA)

FMECA's are used to identify the effect and to assess the criticality of a present malfunction.

(See Figure (2) for worked example).

8.5 Fault Tree Analyse (FTA)

This analysis is in some ways a reverse approach to that of FMECA's insofar as the effect of the problem is and the analysis reviews all of the problems that could cause the effect.

(See Figure (3) for worked example).

9.0 RELIABILITY GROWTH

9.1 Introduction

It is important from a producer's and a customer's point of view to gain viability of Reliability growth i.e. the achievement of contractual reliability values over a period of time.

We can deduce from the previous sections that Product Reliability is not always achieved instantaneously particularly where there is an element of Design and Development involved followed by the use of untimed manufacturing processes and skills but more often by a dedicated approach towards identifying and resolving problems.

9.2 Basic Principles

The basic principle of Reliability Growth Testing is to operate the product in an environment similar to that used by the proposed customer(s), then investigating and eliminating by modification all failures that are thought to be of a repetitive nature i.e. test, analyse, fix, test, analyse, fix etc.

9.3 Factors Affecting Reliability Growth

It has been demonstrated in the past that the success of a Reliability Growth Program depends solely on the efforts put in by the people concerned with the test, analysis, fix procedure.

However, this in turn may be determined by:

a) The nature of the system
b) Complexity
c) Technology
d) Configuration
e) Environment
f) Performance characteristics
g) Cost and time constraints

10.0 PERFORMANCE FEEDBACK

10.1 Introduction

In spite of the various measures that we can put in place to prevent problems we are all human beings and therefore fallible and because of this certain measures have to be put in place to highlight, and the very earliest opportunity, any problem that may occur throughout the project life cycle.

These measures are known as the 'Feedback and corrective action loop'. This simply means that as problems arise a record of the problem has to be made, including the various circumstances prevailing at the time of the occurrence, and necessary decisions made and agreed by all concerned parties on how to prevent a future reoccurrence of the problems on this and similar products.

10.2 Procedure

At the project quality and reliability planning stage the requirements for defect data collection require to be considered, and the various mechanisms for doing so carefully thought out. In principle the following actions require to be considered.

a) When to start defect data collection
The collection of defect data must start as early in the project life cycle as possible and this should be accompanied by the collection of the number of hours run.

b) Responsibilities
Responsibilities should be defined at the outset typically the Reliability Engineer will be responsible for deciding the nature of information that should be collected from the various areas which he/she will process into meaningful statistics.

Other responsibilities that require definition relate to corrective actions that require to be taken as it is pointless to highlight a problem and do nothing about preventing future reoccurrences.

c) Data from customers
In many cases it is difficult to make arrangements with the customer to provide performance feedback information and therefore careful negotiation has to be carried out with the customer in order to clarify objectives and the data collection arrangements.

11.0 RELIABILITY DEMONSTRATIONS Vs RELIABILITY WARRANTIES

11.1 Introduction

Reliability Demonstrations are sometimes required by Customers as a means of demonstrating in an operational environment that contractual reliability requirements have been achieved.

Under the heading of 'Reliability Demonstration' it is often a requirement to also demonstrate 'Availability and Maintainability'.

However, there is an increasing trend from some quarters of the customer base towards not asking for Reliability Demonstrations in favour of a Reliability Warranty.

11.2 Preferred Option

It is not easy to say which is the best from either the customers or the producer's point of view because it is largely dependent on the products that are under consideration and more important the conditions associated with the various contractual arrangements.

However, the following can be said:

a) Configuration and Maintenance
It is unlikely that any producer would offer a Reliability Warranty unless:

- The configuration of the product is frozen
- The organisation offering the warranty is also responsible for maintenance and support

b) Costs
It is likely that Reliability Warranties will always increase basic unit costs.

c) Accountability
Clearly, the producer of a piece of equipment is responsible for the achievement of all contractual conditions including Reliability aspects. However, the acceptance of a Reliability Warranty contract clause does have the effect of making the producer more directly accountable than he may otherwise be.

11.3 Basic Conditions

The basic conditions to be met whether or not the contract demands a 'Reliability Demonstration' or a 'Reliability Warranty' include the following:

a) Design
There is no advantage in submitting a product into a Reliability Demonstration or into service until there is a high degree of assurance that it will achieve the contractual performance requirements.

b) Rules and Regulations
There has to be a clear understanding of the various Rules and Regulations that will be applied to the product in the case of an incident, specifically there has to be guidance as to the determination of whether the incident is attributable or not to the product in order that subsequent discussion with respect to liability can be held on a positive note.

c) Configuration Control
There has to be a definition of the product at all times and this should include the 'Build Status' of all hardware, software etc. elements and should include all agreed configuration changes.

12.0 **DISCUSSION POINTS**

a) Future of Reliability Engineering

b) Organisation

c) Processes

d) Should 'failure' be accepted.

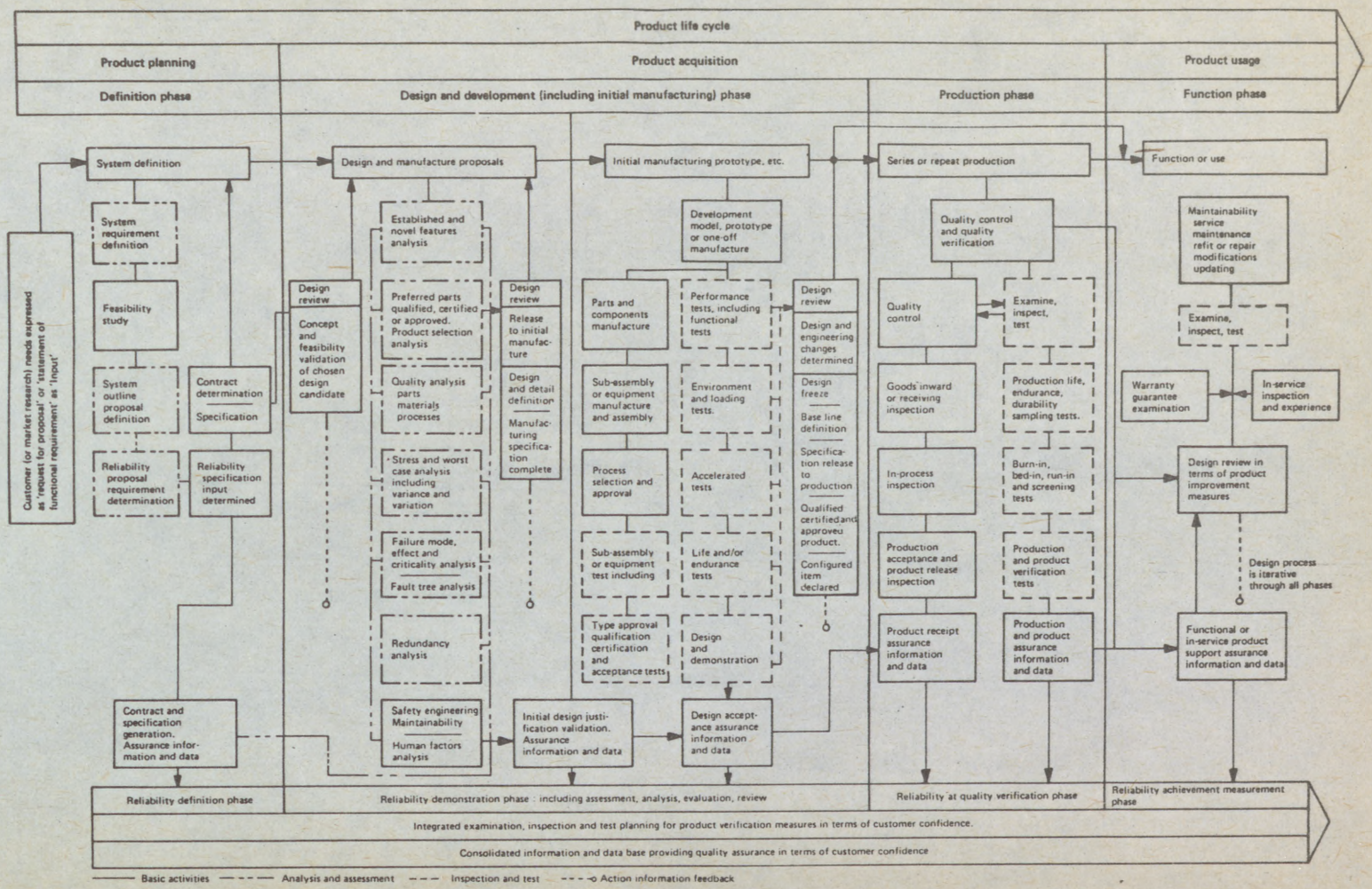

Product life cycle
Product planning
Definition phase
Product acquisition
Design and development (including initial manufacturing) phase
Production phase
Product usage
Function phase
System definition
Design and manufacture proposals
Initial manufacturing prototype, etc.
Series or repeat production
Function or use
Customer (or market research) needs expressed as 'request for proposal' or 'statement of functional requirement' as 'input'
System requirement definition
Feasibility study
System outline proposal definition
Reliability proposal requirement determination
Contract determination
Specification
Reliability specification input determined
Design review
Concept and feasibility validation of chosen design candidate
Established and novel features analysis
Preferred parts qualified, certified or approved. Product selection analysis
Quality analysis parts materials processes
Stress and worst case analysis including variance and variation
Failure mode, effect and criticality analysis
Fault tree analysis
Redundancy analysis
Safety engineering Maintainability
Human factors analysis
Design review
Release to initial manufacture
Design and detail definition
Manufacturing specification complete
Development model, prototype or one-off manufacture
Parts and components manufacture
Performance tests, including functional tests
Sub-assembly or equipment manufacture and assembly
Environment and loading tests.
Process selection and approval
Accelerated tests
Sub-assembly or equipment test including
Life and/or endurance tests
Type approval qualification certification and acceptance tests
Design and demonstration
Design review
Design and engineering changes determined
Design freeze
Base line definition
Specification release to production
Qualified certified and approved product.
Configured item declared
Quality control and quality verification
Quality control
Examine, inspect, test
Goods inward or receiving inspection
Production life, endurance, durability sampling tests.
In-process inspection
Burn-in, bed-in, run-in and screening tests
Production acceptance and product release inspection
Production and product verification tests
Product receipt assurance information and data
Production and product assurance information and data
Maintainability service maintenance refit or repair modifications updating
Examine, inspect, test
Warranty guarantee examination
In-service inspection and experience
Design review in terms of product improvement measures
Design process is iterative through all phases
Functional or in-service product support assurance information and data
Contract and specification generation. Assurance information and data
Initial design justification validation. Assurance information and data
Design acceptance assurance information and data
Reliability definition phase
Reliability demonstration phase : including assessment, analysis, evaluation, review
Reliability at quality verification phase
Reliability achievement measurement phase
Integrated examination, inspection and test planning for product verification measures in terms of customer confidence.
Consolidated information and data base providing quality assurance in terms of customer confidence
Basic activities
Analysis and assessment
Inspection and test
Action information feedback

Figure 1.

FAILURE MODE EFFECT AND CRITICALITY ANALYSIS

a)	Item	a) Cathode Ray Tube
b)	Failure Mode	b) Catastrophic Failure
c)	Failure Effect	c) Equipment unusable
d)	Failure Effect Probability (B)	d) Actual Loss 1.0
e)	Failure Mode Ratio (A) i.e. the fraction of the part failure rate related to the particular failure mode under consideration	e) Estimate 0.8
f)	Part Failure Rate (\)	f) 50 fpmh
g)	Operating Time (t)	g) 40,000 hours
h)	Failure Mode Criticality Cm = BA\t	h) 1.6
i)	System Criticality Cr = (Cm)	i) 1.6 + ...

B Value	Failure Effect
1.0	Actual loss
1.0 to 0.1	Probable loss
0.1 to 0	Possible loss
0	No effect

Figure 2.

FAULT TREE ANALYSIS

For this example we will take a box of matches and will review all of the problems that could prevent a match from igniting.

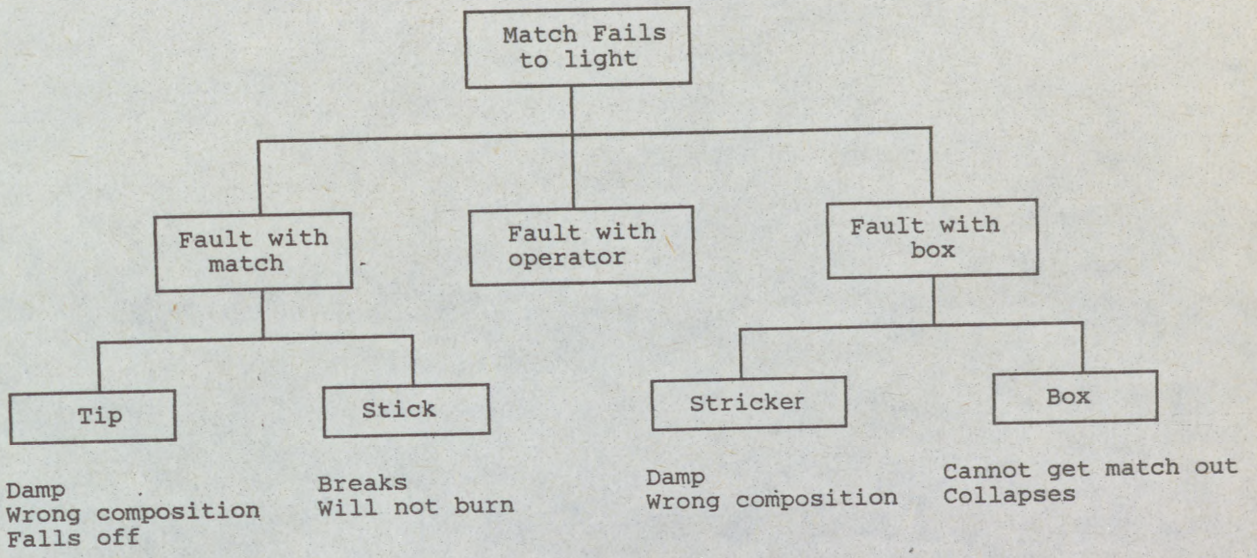


Figure 3.

SYSTEM MODELLING WITH PETRI NETS

Andrea BOBBIO, Istituto Elettrotecnico Nazionale
Galileo Ferraris

Italy

ISTITUTO ELETTROTECNICO NAZIONALE
GALILEO FERRARIS
Strada delle Cacce 91, 10135 Torino

Andrea BOBBIO

SYSTEM MODELLING WITH PETRI NETS

R.T. 361 Settembre 1988

RAPPORTO TECNICO

CONTENTS

SYSTEM MODELLING WITH PETRI NETS

Andrea BOBBIO

Istituto Elettrotecnico Nazionale Galileo Ferraris
Strada delle Cacce 91, 10135 Torino, Italy

ABSTRACT. Petri Nets (PN) are a graphical formalism which is gaining popularity in recent years as a tool for the representation of complex logical interactions (like synchronization, sequentiality, concurrency and conflict) among physical components or activities in a system. This notes are devoted to introduce the formalism of Petri nets with particular emphasis on the application of the methodology in the area of the performance and reliability modelling and analysis of systems. The quantitative analysis of the behaviour of systems in time requires the superposition of a stochastic timing mechanism to the classical representation of PN. Timed Petri nets and, in particular, Stochastic Petri nets (SPN) are the object of the second part of the notes. Finally, some fully developed examples enlighten peculiar aspects which differentiate PNs from other modelling techniques usual in reliability analysis. In few words, the goal of these notes is to show that the proposed methodology based on the PN formalism can be conveniently used as a user-friendly language to represent and evaluate complex stochastic systems.

1. Introduction

Petri Nets (PN) are a graphical tool for the formal description of the flow of activities in complex systems. With respect to other more popular techniques of graphical system representation (like block diagrams or logical trees), PN are particularly suited to represent in a natural way logical interactions among parts or activities in a system. Typical situations that can be modelled by PN are synchronization, sequentiality, concurrency and conflict.

The theory of PN originated from the doctoral thesis of C.A. Petri in 1962 [39]. Since then, the formal language of PN has been developed and used in many theoretical as well as applicative areas. Introductory survey papers can be found in [38,3]. Several textbooks on the subject are also available: [37] (where an extended annotated bibliography is contained) [42,13,44]. An yearly Workshop on "Application and Theory of Petri Nets" is held in Europe (the IX edition of the workshop took place in Venice in June 1988).

The classical PNs do not convey any notion of time; in order to use the PN formalism for the quantitative analysis of the performance and reliability of system versus time, a class of Timed PN (TPN) has been introduced. The time variables associated to the PN can be either deterministic variables (leading to the class of models called deterministic PN), or random variables (leading to the class of models called Stochastic PN - SPN). The bibliography on TPN is not as wide as the one on classical PN, however, an extended

collection of papers and applications can be found in the proceedings of two international workshops specifically devoted to the use of TPN in performance and reliability evaluation [2,1].

The first part (sections 3,4,5,6 and 7) of these lecture notes is aimed at introducing the classical theory of PN, while the second part (sections 8,9,10 and 11) discusses the stochastic timing of a PN with application in the field of reliability modelling and evaluation.

In particular, Section 2 contains the list of symbols. Section 3 defines the primitive elements of the PN and the execution rules by means of which the dynamic properties of the system are described. Section 4 illustrates typical examples of logical interactions among activities modelled by PN, while Section 5 introduces characteristic properties of PNs. Section 6 shows how a PN can be analyzed through the generation of the *reachability tree*, or by means of matrix techniques. Finally, some possible extensions of the modelling capabilities of classical PNs are considered in Section 7.

The second part of the lecture notes is devoted to illustrate how PNs can be conveniently used as a modelling language for the quantitative analysis of the performance and reliability of systems. The use of PN for this purpose requires that the duration of the activities representing the system operations can be specified and measured; therefore, the first step toward the definition of a suitable modelling framework is the insertion of the notion of time in classical PNs. This topic is addressed in Section 8.

Section 9 examines the class of Stochastic PN (SPN) in which the durations of the activities are exponentially distributed random variables. With this assumption, the dynamic behaviour of the PN can be mapped into a continuous-time homogeneous Markov chain. In this way we cast a natural bridge between SPN and Markov models in reliability analysis. With reference to the above models, we discuss the following topics: the generation of the Markov chain associated to the PN, the assignment of marking dependent transition rates and the partition of PN transitions into immediate and timed transitions. Section 10 shows how SPN models can be naturally used to define interesting measures for the characterization of the system behaviour versus time, and how these measures can be computed from the associated Markov chain. Section 11 illustrates some examples of application of the above methodology in reliability analysis. Section 12 briefly introduces the implementation of simulative techniques in the analysis of SPN.

2. List of Symbols

D^-, D^+, D	- Input, Output and Incidence matrix
$\underline{e}_j$	- 0-vector whith entry j equal to 1
$\mathcal{E}$	- Execution sequence
$\mathcal{G}_R$	- Reachability graph
I	- Input function
$HMSPN$	- Homogeneous Markov Stochastic Petri Net
$L = \{\lambda_1, \lambda_2, \ldots, \lambda_{nt}\}$	- Set of firing rates
$M = \{m_1, m_2, \ldots, m_{np}\}$	- Marking
N	- Cardinality of the reachability set (state space)
n_p	- Number of places
n_t	- Number of transitions

O	-	Output function
PN	-	Petri Net
$P = \{p_1, p_2, \ldots, p_{np}\}$	-	Set of places
$\underline{Q}(t)$	-	State probability vector of the associated Markov chain
$\mathcal{R}$	-	Reachability set
SPN	-	Stochastic Petri Net
$T = \{t_1, t_2, \ldots, t_{nt}\}$	-	Set of transitions
$\mathcal{T}_E$	-	Timed execution sequence
TPN	-	Timed Petri Net
$\underline{U}_p$	-	Unitary vector
v	-	Branching probability in a random switch
$\underline{W}_p$	-	Vector of binary (0, 1) entries
$\underline{X}$	-	Integer vector
$\eta_j(t)$	-	Expected number of firings of t_j in $0 - t$
θ_j	-	Random firing time associated to t_j
$\lambda, \mu, \gamma, \rho$	-	Firing rates
Λ	-	Transition rate matrix of the associated Markov chain
ϕ	-	Mean passage time
τ_j	-	Epoch of firing of $t_{(j)}$
$\psi(t)$	-	Expected time spent in a marking in $0 - t$
ω	-	Infinite reproducibility in the reachability tree

3. The Primitive Elements of a Petri Net

For definitions and notation we refer in general to [37]. A Marked PN is a quintuple (P, T, I, O, M), where:

- $P = \{p_1, p_2, \ldots, p_{np}\}$ is the set of n_p places (drawn as circles in the graphical representation);
- $T = \{t_1, t_2, \ldots, t_{nt}\}$ is the set of n_t transitions (drawn as bars);
- I is the transition input relation and is represented by means of arcs directed from places to transitions;
- O is the transition output relation and is represented by means of arcs directed from transitions to places;
- $M = \{m_1, m_2, \ldots, m_{np}\}$ is the marking. The generic entry m_i is the number of tokens (drawn as black dots) in place p_i in marking M.

The graphical structure of a PN is a bipartite directed graph: the nodes belong to two different classes (places and transitions) and the edges (arcs) are allowed to connect only nodes of different classes (multiple arcs are possible in the definition of the I and O relations [37]). Figure 1 is a PN [3].

The dynamics of a PN is obtained by moving the tokens in the places by means of the following execution rules:

- A transition is enabled in a marking M if all its input places carry at least one token;
- an enabled transition fires by removing one token per arc from each input place and adding one token per arc to each output place.

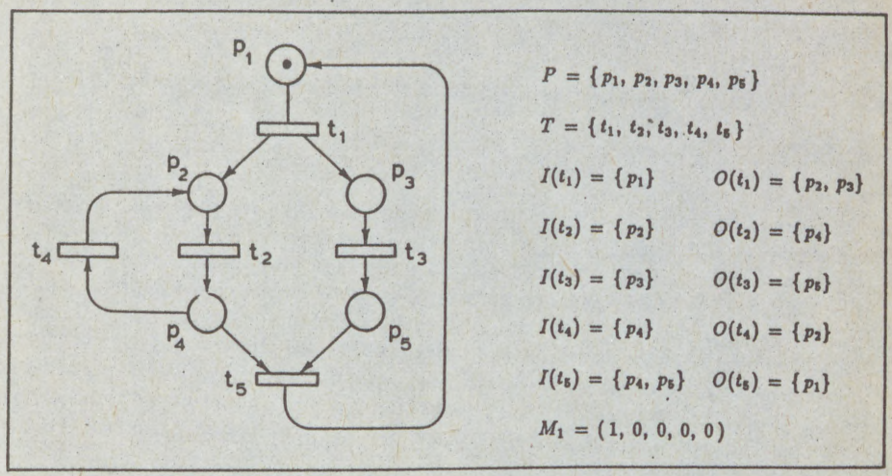


Figure 1 - A PN graph with Input and Output functions.

Given an initial marking M_1, the reachability set $\mathcal{R}(M_1)$ is the set of all the markings that can be obtained by repeated application of the above rules.

More formally we can say that t_k is enabled in marking M if:

$$\text{for any} \quad p_i \in I(t_k) \quad , \quad m_i \geq 1$$

Marking M', obtained from M by firing t_k, is said to be *immediately reachable* from M, and the firing operation is denoted by the symbol $(M - t_k \rightarrow M')$. The token count in M' is pictorially represented in Figure 2, and is given by the following relationship:

$$M'(p_i) = \begin{cases} M(p_i) + 1 & \text{if} \quad p_i \in O(t_k)\,, \quad p_i \notin I(t_k) \\ M(p_i) - 1 & \text{if} \quad p_i \notin O(t_k)\,, \quad p_i \in I(t_k) \\ M(p_i) & \text{otherwise} \end{cases}$$

Let us examine the generation of the reachability set of the PN of Figure 1 given the initial marking $M_1 = (1,0,0,0,0)$. In M_1 the only enabled transition is t_1; firing

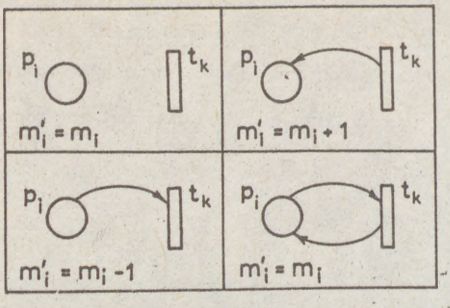


Figure 2 - Token number modification in place p_i subsequent to the firing of transition t_k.

of t_1 removes the token from p_1 and puts a token both in p_2 and p_3 producing the new marking $M_2 = (0,1,1,0,0)$. In M_2 the transitions t_2 and t_3 are both enabled and can fire concurrently. Firing of t_3 leads to $M_3 = (0,1,0,0,1)$ and subsequent firing of t_2 leads to $M_4 = (0,0,0,1,1)$. In M_4 transitions t_4 and t_5 are both enabled, but the firing of either disables the other; the two transitions are in conflict. Firing of t_4 in M_4 produces marking M_3, while firing t_5 in M_4 produces the intial marking M_1. Note that a different firing sequence can be activated from marking M_2, by letting t_2 firing first and obtaining marking $M_5 = (0,0,1,1,0)$. With this, all the possible firing sequences have been examined, and the reachability set $\mathcal{R}(M_1)$ of the net of Figure 1 turns out to contain 5 elements M_1, M_2, M_3, M_4 and M_5.

4. Petri Nets and the Modelling of Systems

PN used for modelling real systems are sometimes referred to as *Condition/Events* nets. Places identify the conditions of the parts of the system (working, idle, queueing, failed), and transitions describe the passage from one condition to another (end of a task, failure, repair ...). An event occurs (a transition fires) when all the conditions are satisfied (input places are marked) and give concession to the event. Occurrence of the event modifies in whole or in part the status of the conditions (marking). The number of tokens in a place can be used to identify the number of resources lying in the condition denoted by that place. The following examples illustrate typical situations of interaction of activities arising in system modelling.

4.1. CONCURRENCY (OR PARALLELISM)

In the PN of Figure 3 transitions t_1 and t_2 are enabled simultaneously; the firing of one of them does not modify the state of the other. The activities modelled by the two transitions run concurrently. In reliability modelling, the PN of Figure 3 can represent two components C_1 and C_2 in parallel redundancy; in this case, places p_1 and p_3 represent the working

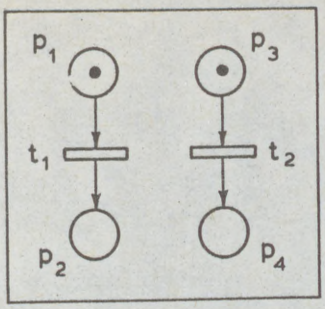


Figure 3 - PN modelling two parallel activities.

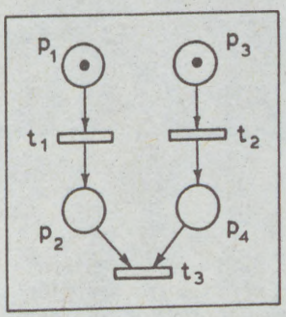


Figure 4 - A PN modelling two parallel activities with synchronization.

condition, p_2 and p_4 the failed condition and t_1 and t_2 the event of failure of C_1 and C_2 respectively.

4.2. SYNCHRONIZATION

In Figure 3 the activities modelled by t_1 and t_2 run concurrently; however, if they represent routines of a parallel program, both should be terminated before the program execution can proceed. The synchronization activity is modelled in Figure 4 by means of transition t_3 whose firing requires a token both in p_2 and p_4.

4.3. LIMITED RESOURCES

A typical factor influencing the performance of distributed systems (multiprocessor sys-

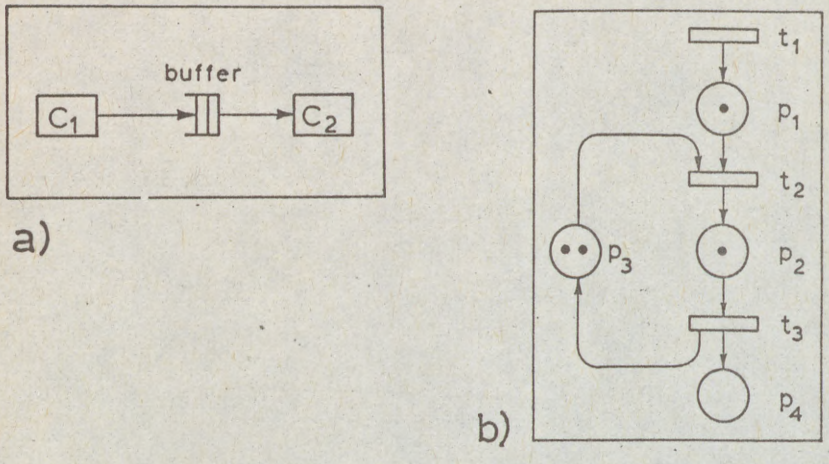


Figure 5 - Block diagram and PN of a buffer with finite size.

tems, flexible manufacturing systems and so on) is the limited number of available resources. Exhaustion of the resources prevents the activities to proceed and blocks the system. Modelling and analysing systems with blocking is a difficult task in almost all modelling frameworks [15,46]. A PN representation of a buffer with limited size is shown in Figure 5b (Figure 5a shows the corresponding block diagram representation). Place p_3 models the number of free buffer positions whereas p_2 the number of filled positions; note that the sum of tokens in p_2 and p_3 is constant and models the total number of available buffer positions (three positions in the figure). Transition t_2 models the filling of one buffer position and can fire if a position free (at least one token in p_3) exists and a task is available to be stored (a token in p_1). Transition t_3 is enabled when at least one buffer position is filled, and firing of t_3 moves one token from p_2 to p_3.

4.4. SEQUENTIALITY (THE PRODUCER/CONSUMER PROBLEM)

A producer produces objects that are put into a buffer from which can be removed and consumed by a consumer. The consuming process must be in sequence with respect to the production process. The PN solution to this problem is reported in Figure 6. A token in p_1 means that the producer is ready to produce. By firing t_1 and t_2 an object is produced (a token is put in the buffer p_5) and the producer is ready again. If the consumer is ready to consume (token in p_3) and an object is in the buffer, transition t_3 can fire removing one token from p_5.

In the PN of Figure 6 the production and the accumulation of objects in the buffer is unbounded. A more realistic situation is obtained by considering a buffer of limited capacity (as in 4.3). The corresponding PN is reported in Figure 7. Place p_6 models the

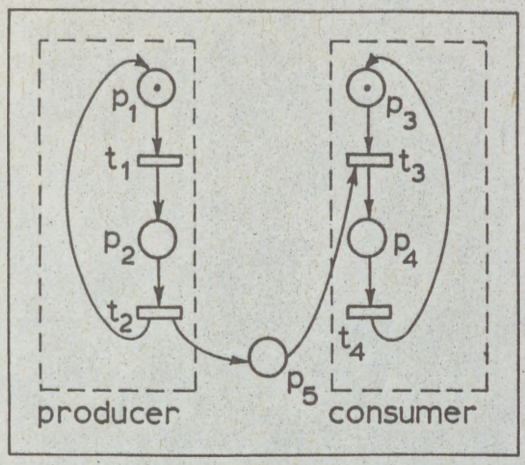


Figure 6 - The producer/consumer problem with unbounded buffer.

free buffer positions and place p_5 the filled buffer positions; the number of tokens in p_5 and p_6 is constant and represents the total available buffer positions. If a single token is assigned to p_6 in the initial marking, we model the situation in which the producer cannot further produce until the consumer has consumed the object in the buffer (a strictly sequential ordering of activities).

4.5. MUTUAL EXCLUSION (CONFLICT)

Two resources C_1 and C_2 are allowed to work in parallel, but are connected to a shared resource C_s that cannot be accessed by C_1 and C_2 simultaneously (block diagram in Figure 8a). The corresponding PN is in Figure 8b. Places p_1 and p_5 represent C_1 and C_2 working independently; p_2 and p_6 represent C_1 and C_2 requesting access to C_s; p_3 and p_7 represent C_s busy with C_1 and C_2 respectively. Place p_4 determines which resource can actually access C_s, and prevents places p_3 and p_7 to be marked at the same time; in fact when p_2 and p_6 are both marked, transitions t_2 and t_5 are in conflict. Firing of one of them disables the other. Firing of t_3 or t_6 models the release of the common resource (token back in p_4) and the return to the working condition.

5. Properties of Petri Nets

We enumerate different properties which allow us to classify the primitive elements of a PN or the PN as a whole.

5.1. LIVENESS

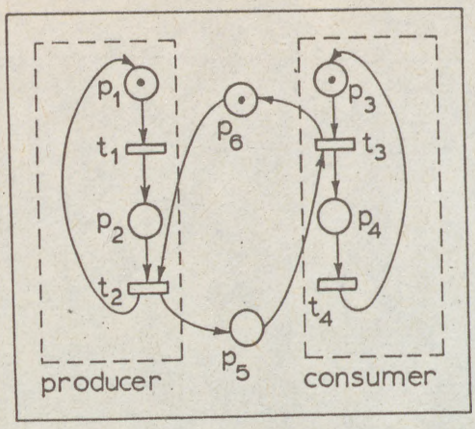


Figure 7 - The producer/consumer problem with finite buffer.

A transition is *potentially firable* in M if there exists a sequence of transition firings which leads to a marking in which the transition is enabled. A transition is *live* if it is potentially firable in any marking of $\mathcal{R}(M_1)$. A transition is *dead* in M if it is not potentially firable; if the PN enters marking M the dead transition cannot fire any more.

5.2. SAFENESS

A place is *safe* if the token count does not exceed 1 in any marking of $\mathcal{R}(M_1)$. A PN is safe if each place is safe. The PNs of Figures 1, 3 and 8b are safe.

5.3. BOUNDEDNESS

A simple generalization of safeness is the concept of boundedness. A place is bounded with bound k, if the token count does not exceed k in any marking of $\mathcal{R}(M_1)$. A PN is k-bounded if each place is k-bounded. The PN of Figure 7 is k-bounded where k is the number of buffer positions. On the contrary, the PN of Figure 6 is unbounded.

5.4. CONSERVATION

A PN is strictly conservative if the total number of tokens is constant in each marking of $\mathcal{R}(M_1)$. The PN of Figure 7 is k-bounded and strictly conservative, while the PN of Figure 8b is safe but not strictly conservative. A subset of places form a *place-invariant* [31] if it is strictly conservative. In the net of Figure 8b the subsets $\{p_1, p_2, p_3\}$, $\{p_5, p_6, p_7\}$ and $\{p_3, p_4, p_7\}$ are place-invariants.

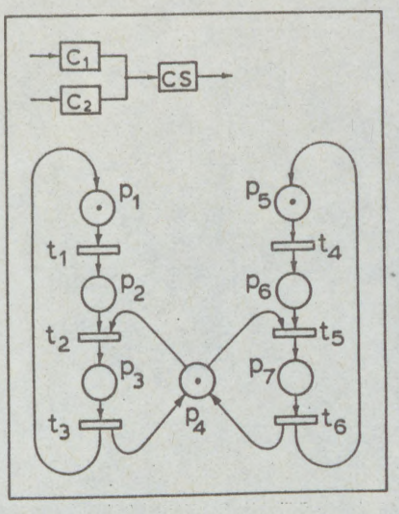


Figure 8 - The mutual exclusion problem: two parallel resources with a common shared block.

6. Analysis Techniques

The success of any model depends on two factors: its modelling power and its decision power. Modelling power refers to the ability to correctly represent the system to be modelled; decision power refers to the ability to analyze the model and determine properties of the modelled system. The modelling power of PN has been examined in the previous sections, and in this section we take into consideration the analysis techniques of PNs.

6.1. THE REACHABILITY TREE AND REACHABILITY GRAPH

The reachability set $\mathcal{R}(M_1)$ of a PN is generated by means of the reachability tree. The initial marking M_1 is the root of the reachability tree. Starting from the root we search for all the enabled transitions; the firing of an enabled transition produces a new marking which is represented as a new leaf in the tree, from which the procedure is iterated.

By properly identifying the frontier nodes of the tree, the generation of the reachability tree involves a finite number of steps [37], even if the PN is unbounded. Let us introduce three kinds of frontier nodes:

- terminal (dead) nodes: nodes in which no transitions are enabled;
- duplicate nodes: nodes which have been already generated in the tree;

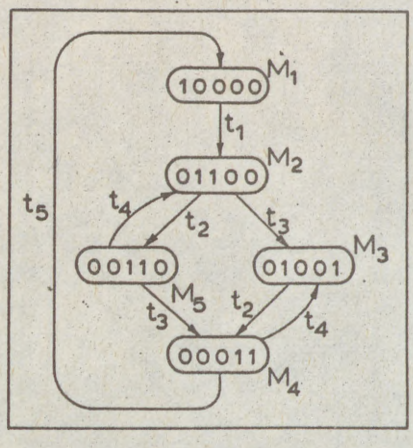


Figure 9 - Reachability graph $\mathcal{G}_R(M_1)$ for the net of Figure 1.

- infinitely reproducible nodes. A marking M'' is an infinitely reproducible node if $M'' \geq M'$ ($m''_i \geq m'_i$, $i = 1, 2, \ldots, n_p$) for some M' already generated in the tree. Because of the stated relation, the transition sequence from which M'' has been generated starting from M' is surely firable in M''. Thus, the sequence $M' \rightarrow M''$ can be reproduced infinitely often, so that the token count in the places for which $m''_i \geq m'_i$ can increase indefinitely. We represent the arbitrarily large number of tokens which results from infinitely reproducible nodes by defining a special symbol ω with the following properties:

$$\begin{aligned} \omega + a &= \omega \\ \omega - a &= \omega \\ a &< \omega \end{aligned}$$

for any positive constant a.

By allowing ω to be a legal symbol in the reachability tree specification, it can be shown that the generation of the reachability tree involves always a finite search algorithm [37]. If the generation of the reachability tree terminates without arriving to infinitely reproducible nodes, the PN is bounded. In this case the reachability set is finite and can be represented as a labelled directed graph whose vertices are the elements of $\mathcal{R}(M_1)$ and such that for each possible transition firing $M_i - t_k \rightarrow M_j$ there exists an arc (i, j) labelled k. The reachability graph associated to a reachability set $\mathcal{R}(M_1)$ will be denoted by $\mathcal{G}_R(M_1)$.

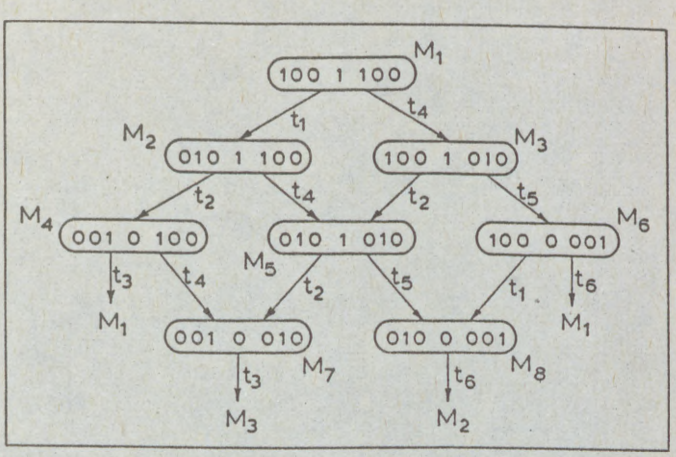


Figure 10 - Reachability graph $\mathcal{G}_R(M_1)$ for the net of Figure 8.

Figure 9 shows the reachability graph of the PN of Figure 1 with initial marking $M_1 = (1,0,0,0,0)$, as discussed in Section 3. Figure 10 shows the reachability graph for the mutual exclusion problem of Figure 8, with initial marking $M_1 = (1,0,0,1,1,0,0)$.

In Figure 11 we have reported the reachability tree of the PN of Figure 6; since the net is unbounded, in order to keep the generation algorithm finite, the symbol ω has been introduced.

If a PN has a finite $\mathcal{R}(M_1)$ all the properties of the net (safeness, liveness, etc..) can be analyzed by inspection of the reachability graph. If the net is unbounded the finite reachability tree representation, by means of the symbol ω, can be an imperfect description of the net (it is possible to find PNs with different properties and behaviours that cannot be distinguished through the reachability tree, due to incomplete information carried by ω [37]).

6.2. MATRIX ANALYSIS

The input and output functions of a PN can be equivalently defined using a matrix notation. Let D^- denotes the input matrix. D^- is a $(n_t \times n_p)$ matrix, whose generic element d^-_{ij} is equal to the number of arcs connecting place p_j with transition t_i. Similarly we define the output matrix D^+ as a $(n_t \times n_p)$ matrix, whose generic element d^+_{ij} is equal to the number of arcs connecting transition t_i with place p_j. The *incidence matrix* D is defined by the following relation:

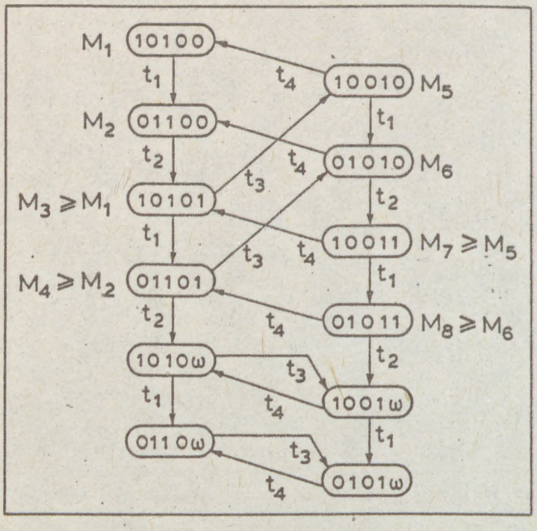


Figure 11 - Generation of the reachability tree for the PN of Figure 6 with unbounded buffer.

$$D = D^+ - D^- \tag{1}$$

The matrices D^-, D^+ and D for the PN of Figure 8b are reported in the following:

$$D^- = \begin{array}{c|ccccccc} & p_1 & p_2 & p_3 & p_4 & p_5 & p_6 & p_7 \\ \hline t_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ t_2 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ t_3 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ t_4 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ t_5 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ t_6 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}$$

$$D^+ = \begin{array}{c|ccccccc} & p_1 & p_2 & p_3 & p_4 & p_5 & p_6 & p_7 \\ \hline t_1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ t_2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ t_3 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ t_4 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ t_5 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ t_6 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{array}$$

$$D = \begin{array}{c|ccccccc} & p_1 & p_2 & p_3 & p_4 & p_5 & p_6 & p_7 \\ \hline t_1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ t_2 & 0 & -1 & 1 & -1 & 0 & 0 & 0 \\ t_3 & 1 & 0 & -1 & 1 & 0 & 0 & 0 \\ t_4 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ t_5 & 0 & 0 & 0 & -1 & 0 & -1 & 1 \\ t_6 & 0 & 0 & 0 & 1 & 1 & 0 & -1 \end{array} \qquad (2)$$

Let us further introduce the vector $\underline{e}_j$ which is a n_t-dimensional row vector with all the entries equal to 0 except entry j equal to 1. With this notation the execution rules of a PN becomes:

- a transition t_j is enabled in marking M iff $M \geq \underline{e}_j D^-$ (note that $\underline{e}_j D^-$ is the $j-th$ row of D^-);
- firing of t_j in M produces a marking M' given by:

$$M' = M - \underline{e}_j D^- + \underline{e}_j D^+ = M + \underline{e}_j D \qquad (3)$$

From the previous definitions follows that, given a PN with initial marking M_1 and a firing sequence $t_i \rightarrow t_j \rightarrow t_k \rightarrow t_j \rightarrow t_i$, the marking obtained at the end of the sequence is given by the following matrix equation:

$$M_{fin} = M_1 + (\underline{e}_i + \underline{e}_j + \underline{e}_k + \underline{e}_j + \underline{e}_i)\, D \qquad (4)$$

By means of the matrix representation, the following properties of PN can be inspected.

6.2.1. Reachability - A marking M' is reachable from M if an integer vector $\underline{X}$ exists such that (see equation 3):

$$M' = M + \underline{X} D \qquad (5)$$

Equation (5) provides a necessary but not sufficient condition; all markings reachable from M are solution of equation (5) but not viceversa; for any integer vector $\underline{X}$ a solution to

equation (5) exists, but the transition firing sequence represented by $\underline{X}$ can be non-firable. Furthermore, note that the solution of (5) is not affected by the order of transition firings (but only by the number), while the semantics of the net is strongly affected by the order: changing the order a legal sequence can become non-firable.

6.2.2 Conservation - Given a conservative PN, and a n_p-dimensional column vector $\underline{U}_p^T$ with all the entries equal to one, for any marking $M' \in \mathcal{R}(M_1)$ the following relation should hold:

$$M_1 \underline{U}_p^T = M' \underline{U}_p^T \tag{6}$$

Thus, from equation (5):

$$M_1 \underline{U}_p^T = M_1 \underline{U}_p^T + \underline{X} D \underline{U}_p^T \tag{7}$$

since (5) must be satisfied for any $\underline{X}$, it follows:

$$D \underline{U}_p^T = 0 \tag{8}$$

Equation (8) is a necessary and sufficient condition for conservation.

6.2.3. Place Invariant - Let $\underline{W}_p$ be a vector of binary entries (either 0 or 1); we find all vectors $\underline{W}_p$ for which [31]:

$$D \underline{W}_p^T = 0 \tag{9}$$

the places p_i $(i = 1, 2, \ldots, n_p)$ for which $w_i = 1$, form a *place invariant* (a conservative subset of places). With reference to the incidence matrix D of Equation (2), it is easily verified that the following vectors are solution of Equation (9):

$$\begin{aligned} \underline{W}_p^{(1)} &= [1\,1\,1\,0\,0\,0\,0] \\ \underline{W}_p^{(2)} &= [0\,0\,0\,0\,1\,1\,1] \\ \underline{W}_p^{(3)} &= [0\,0\,1\,1\,0\,0\,1] \end{aligned}$$

and therefore, the subsets $\{p_1, p_2, p_3\}$ $\{p_5, p_6, p_7\}$ and $\{p_3, p_4, p_7\}$ are place invariant for the PN of Figure 8b.

7. Extensions

In the use of PN for modelling real systems several authors have found convenient to introduce special constructs either for making the model representation more compact in a given application or for extending the modelling power of the PN formalism. The extensions more often encountered in the literature (and that will be used in the sequel), have been proposed in response to difficulties in modelling priority disciplines by PN. All the extensions mentioned in the sequel are equivalent from the point of view of the modelling power, thus their use depends on the easiness or convenience of the implementation [16].

7.1. INHIBITOR ARCS

An inhibitor arc from place p_j to transition t_k modifies the enabling rules in the sense that the transition can fires only if place p_j does not contain tokens. The inhibition function is usually represented by circle-headed arcs, as in Figure 12 where transition t_k can fire iff p_i contains at least one tokens, but no tokens are present in p_j.

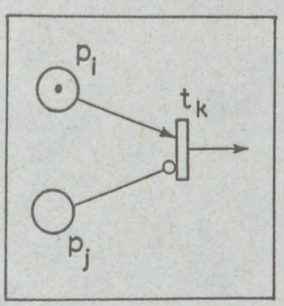


Figure 12 - Inhibitor arc.

In the mutual exclusion problem of Figure 8, the standard PN language does not provide any means to establish precedence rules in the case both resources C_1 and C_2 are simultaneoulsy requesting access to the common resource C_s (places p_2 and p_5 simultaneously marked). With the insertion of an inhibitor arc from place p_2 to transition t_5 (Figure 13), we model the situation in which, as a conflict arises between C_1 and C_2, C_1 has always the precedence, and blocks (inhibits) C_2 until the common resource is released.

With respect to the reachability graph $\mathcal{G}_R(M_1)$ of the original PN of Figure 8 (reported in Figure 10), the reachability graph of the modified PN of Figure 13 is such that from marking M_5 only transition t_2 can fire while t_5 is inhibited.

7.2. PRIORITY LEVELS

An alternative, but equivalent way to model the same features considered with the introduction of inhibitor arcs, is obtained by attaching to each PN transition a priority level. The standard execution rules are modified in the sense that, among all the transitions enabled in a given marking, only those with associated highest priority level are allowed to fire. In Figure 13 exactly the same precedence policy can be modelled by attaching to transition t_2 a priority level greater than the one attached to t_5. In marking M_5 (see Figure 10) in which both transitions are enabled, only t_2 can fire.

7.3. CONDITIONING FUNCTIONS

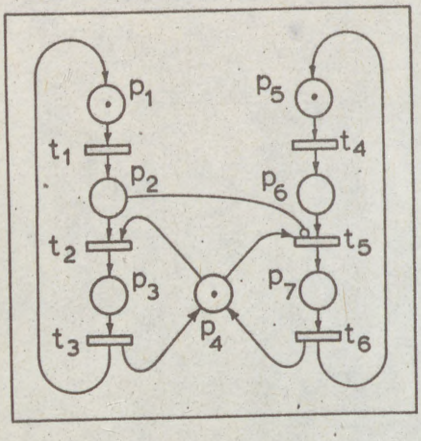


Figure 13 - The mutual exclusion problem of Figure 8, with precedence on the resource C_1.

More complex logical interactions between primitive elements of a PN can be considered by introducing logical *conditioning functions* [20]. Given a marking M, a PN transition is enabled if, beside the normal enabling requirements (including inhibitor arcs and priorities), the conditioning function is *true*. The conditioning functions can be very effective in reducing the graphical complexity of a PN, even if they do not extend the modelling power with respect to inhibitor arcs or priority levels.

7.4. HIGH LEVEL PETRI NETS

In the PN models discussed so far, the individual tokens are indistiguishable. The semantics of the model does not allow to follow the behaviour of an individual token through the net. To overcome this limitation a new class of models has been proposed and discussed. The common characteristic of these models, usually referred to as *high level* PN, is that the position of any single token can be tracked in the PN. Two labelling techniques have been originally proposed: the technique of colouring tokens (coloured PN introduced by Jensen [28]) and the technique of assigning to each token a predicate (*Predicate/Transition* net introduced by Genrich and Lautenbach [24]). However, this class is not further dealt with in the present notes.

8. Timed Petri Nets

An *execution sequence* $\mathcal{E}$ in a marked PN, is a sequence of legal markings obtained by firing a sequence of enabled transitions:

$$\mathcal{E} = \{(M_{(1)}, t_{(1)}); (M_{(2)}, t_{(2)}); \ldots; (M_{(j)}, t_{(j)}); \ldots\}$$

An execution sequence $\mathcal{E}$ can be viewed as a connected path in the reachability graph $\mathcal{G}_R(M_1)$ of the net.

A timed execution sequence $\mathcal{T}_E$ of a marked PN with intial marking $M_{(1)}$, is an execution sequence $\mathcal{E}$ augmented by a non-decreasing sequence of real values representing the epochs of firing of each transition, such that consecutive transitions $(t_{(j)}; t_{(j+1)})$ in $\mathcal{E}$ correspond to ordered epochs $\tau_j \leq \tau_{j+1}$ in $\mathcal{T}_E$. Thus formally [23,5]:

$$\mathcal{T}_E = \{(M_{(1)}, t_{(1)}, \tau_1); (M_{(2)}, t_{(2)}, \tau_2); \ldots; (M_{(j)}, t_{(j)}, \tau_j); \ldots\}$$

The time interval $\tau_j - \tau_{j+1}$ between consecutive epochs represents the period that the PN sojourns in marking $M_{(j)}$. In the sequel we always assume as initial epoch $\tau_1 = 0$.

> **Definition** - *A Timed PN (TPN) is a marked PN in which a set of specifications are provided and a set of rules are defined such that to each legal execution sequence $\mathcal{E}$ a timed execution sequence $\mathcal{T}_E$ can be univocally associated.*

A variety of timing mechanisms have been proposed in the literature. The distinguishing features of the timing mechanisms are whether the duration of the events is modelled by deterministic variables or random variables, and whether the time is associated to the PN places, transitions or tokens.

Earlier work in timed PN with deterministic timing can be found in [32,43,40,47]. Application of deterministic-PN models are available in different areas, like: communication protocols, performance evaluation, manufacturing. However, in the reliability area stochastic modelling is more appropriate, and therefore we will consider in the sequel only TPN in which the timing mechanism is stochastic; we will refer to this class of models as Stochastic PN (SPN).

SPN were initially proposed in two doctoral thesis [36,35]. In these models, interpreting PN as Condition/Event nets, time was naturally associated with activities that induce state changes, hence with the delay incurred before firing transitions. Although other possibilities have been explored, the choice of associating time with PN transitions is the most common in the literature, and is the only considered in the present notes.

When the random variables associated to PN transitions are exponentially distributed, the dynamic behaviour of the PN can be mapped into a time-continuous homogeneous Markov chain with state space isomorphic to the reachability graph of the PN. This case will be considered in details in the following sections.

Extensions to cover the case of generally distributed transition firing times have been considered in a number of papers [36,21,5,23,26,4]. Releasing the memoryless property of the exponential distribution, in order to univocally associate to each execution sequence $\mathcal{E}$ a timed execution sequence $\mathcal{T}_E$ the concept of SPN *execution policy* needs to be introduced

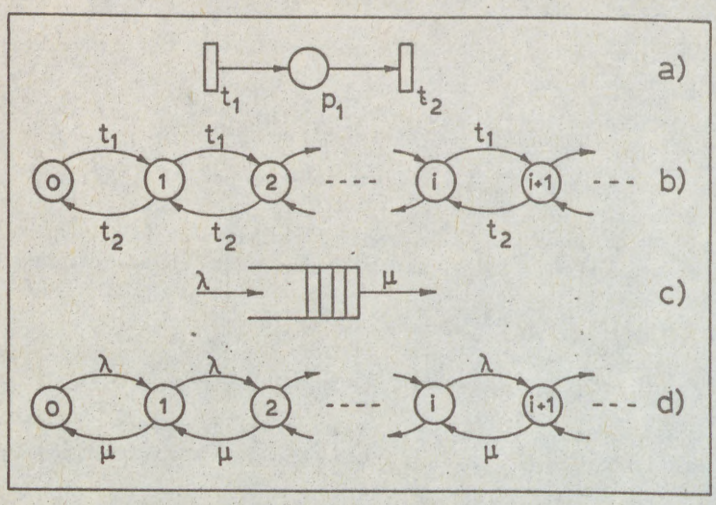


Figure 14 - The $M/M/1$ queue: a) The PN representation;
b) The reachability graph; c) The block diagram representation;
d) The corresponding Markov transition graph.

[5,4]. The execution policy consists of two parts: the way in which a transition is selected to fire among those enabled in a given marking, and the way in which the time spent is recovered after a transition firing. However, due to the complexity of the semantics of the SPN models, and of the associated stochastic process (both aspects are strictly dependent on the definition of the execution policy), this generalization is no further considered in this paper.

9. Homogeneous Markov SPN (HMSPN)

Let us suppose that the activity modelled by a PN transition takes an exponentially distributed random amount of time to complete once initiated. This means that an exponentially distributed random variable θ_j with parameter $\lambda_j(M)$ is associated to each PN transition t_j. The firing of an enabled transition t_j in marking M becomes a random event which occurs with a time-independent (but possibly marking dependent) firing rate $\lambda_j(M)$. Therefore, knowing the transitions enabled in a given marking and the associated firing rates, we can univocally generate the stochastically timed sequence $\mathcal{T}_E$ from each execution sequence $\mathcal{E}$. In other words, the reachability graph $\mathcal{G}_R(M_1)$ of a marked PN can be univocally mapped into a discrete-state continuous time homogeneous Markov chain, by letting each marking of $\mathcal{G}_R(M_1)$ correspond to a state in the Markov chain, and by substituting the label of the PN transition in each edge of $\mathcal{G}_R(M_1)$ with the firing rate of

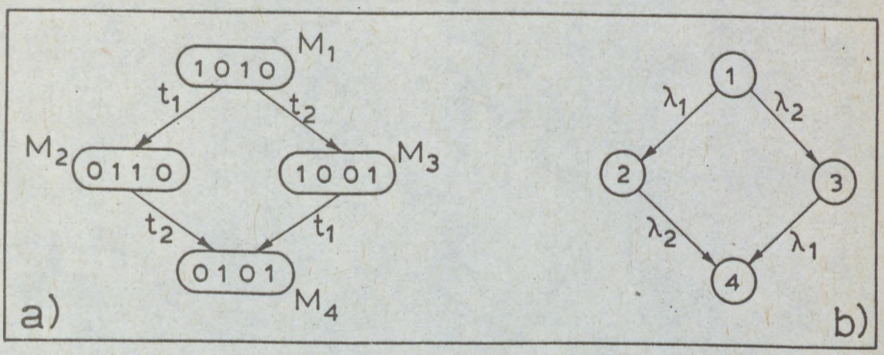


Figure 15 - The reachability graph a) and the corresponding Markov chain b) of the SPN of Figure 3.

the corresponding transition. With this definition we can speak indifferently of *marking* M_i or *state* i.

Example 1 - The M/M/1 queue. Consider the PN of Figure 14a) where it is intended that a transition with no input places is always enabled. The corresponding reachability graph is reported in Figure 14b). The label inside each state is the marking, i.e. the number of tokens in place p_1. Firing of t_1 increases the token count by 1, while firing of t_2 decreases the token count by 1. By associating to transition t_1 the arrival rate λ and to t_2 the service rate μ, the PN of Figure 14a) models the *M/M/1* [29] queueing system. The usual block diagram representation is given in Figure 14c) and the corresponding Markov transition graph is given in Figure 14d). This example is also intended to show how the PN language is suitable to represent queueing systems or queueing networks.

Example 2 - Let the PN of Figure 3 denote the failure process of two components in parallel redundancy; t_1 is the event of failure of component 1 to which a failure rate λ_1 is assigned. Similarly we assigne to t_2 the failure rate λ_2 of component 2. Figure 15a) shows the reachability graph of the net and Figure 15b) the associated Markov chain representing the dynamic behaviour of the net in time.

The probability of the original PN of being in marking M_4 at time t where both components are failed can be computed as the probability of being in state 4 at time t in the corresponding Markov chain.

Example 3 - The reachability graph of the PN of Figure 8 is reported in Figure 10. If all the PN transitions are assigned time-independent firing rates, the reachability graph of Figure 10 is mapped into the Markov chain of Figure 16.

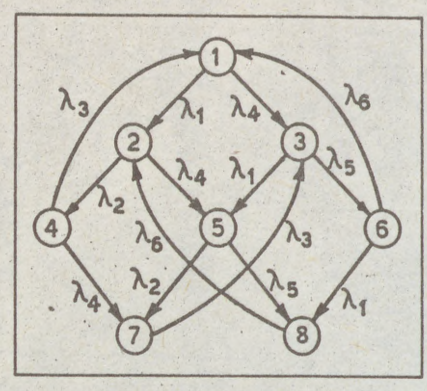


Figure 16 - Markov chain corresponding to the reachability graph of Figure 10.

9.1. FORMAL DEFINITION OF THE MODEL

The Homogenous Markov SPN (HMSPN) is a six-tuple:

$$HMSPN = (P, T, I, O, M, L)$$

where P, T, I, O, M have the same meaning introduced in Section 3, and $L = \{\lambda_1(M), \lambda_2(M), \ldots, \lambda_{nt}(M)\}$ is a set of n_t non-negative real numbers representing the (marking dependent) firing rates of the exponential random variables associated to each PN-transition.

The knowledge of the reachability graph allows us to automatically generate the transition rate matrix Λ of the associated homogeneous Markov chain. Λ is a $N \times N$ matrix, where N is the cardinality of the reachability set $\mathcal{R}(M_1)$.

Let us define $\underline{Q}(t)$ a N-dimensional state probability vector, whose generic entry $q_i(t)$ is the probability of being in state i $(i = 1, 2, ..., N)$ at time t in the associated Markov chain. $\underline{Q}(t)$ is the solution of the standard Markov linear differential equation:

$$\frac{d\,\underline{Q}(t)}{d\,t} = \Lambda\,\underline{Q}(t) \tag{10}$$

with initial condition $\underline{Q}(0) = [1, 0, 0, ..., 0]^T$. If the steady state probability vector $\underline{Q}(\infty)$ of the Markov chain exists, it can be calculated from the equation:

$$\Lambda\,\underline{Q}(\infty) = \underline{0} \qquad \text{with} \qquad \sum_{i=1}^{N} q_i(\infty) = 1 \tag{11}$$

The numerical techniques for the solution of Equations (10) and (11) are outside the scope of the present notes. For a recent survey on methods and techniques for solving equation (10) see [41].

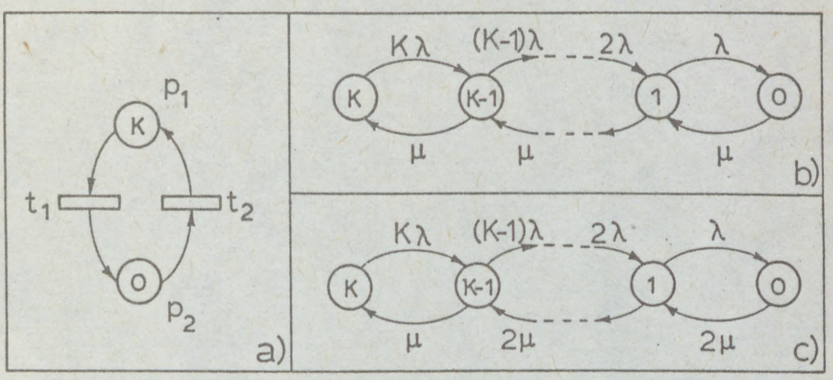


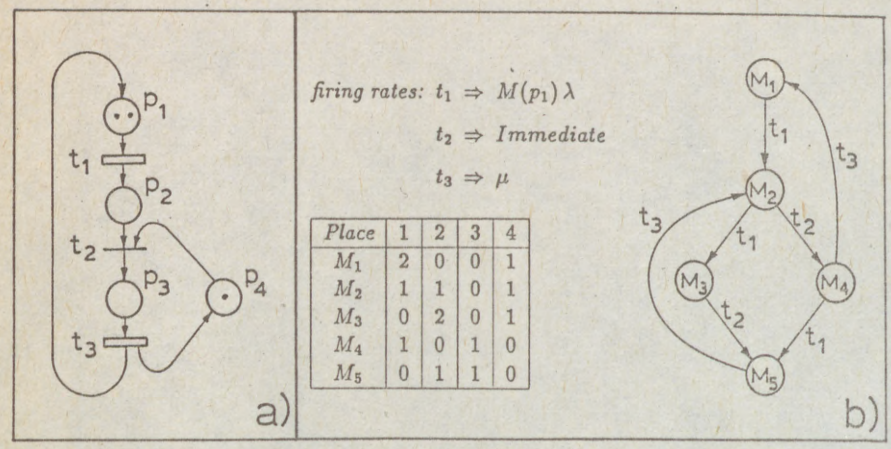
Figure 17 - a) PN modelling K identical parallel components;
b) The associated Markov chain with a single repairman;
c) The associated Markov chain with 2 repairmen.

9.2. MARKING DEPENDENT FIRING RATES

In the formal definition of the HMSPN model the firing rates associated to each transition have been considered as marking dependent. This possibility increases the flexibility of the model and is often used to make the models more compact in the case of the presence of multiple identical resources.

Example 4 - The PN of Figure 17a) has the following physical meaning: place p_1 represents operation; place p_2 non operation; transition t_1 failure and transition t_2 repair. Suppose we have K identical components in parallel redundancy each one with failure rate λ. We can model the system operation by the PN of Figure 17a) with initial marking $M_1 = (K, 0)$ and associating to transition t_1 the marking dependent transition rate $\lambda_{t_1}(M_x) = m_{1x}\lambda$, where m_{1x} is the number of tokens in place p_1 in marking M_x.

Moreover, we can easily model various repair policies: the single repairman policy is modelled by assigning to transition t_2 the repair rate μ. In this case, the Markov chain corresponding to the PN of Figure 17a) is reported in Figure 17b). The independent repair policy can be modelled by assigning to transition t_2 the marking dependent firing rate $\mu_{t_2}(M_x) = m_{2x}\mu$ (as many repairmen as failed components m_{2x}). The case of two repairmen can be modelled by means of more complex logical assignment to the firing rate $\mu_{t_2}(M_x)$ of transition t_2:



Place	1	2	3	4
M_1	2	0	0	1
M_2	1	1	0	1
M_3	0	2	0	1
M_4	1	0	1	0
M_5	0	1	1	0

Figure 18 - a) Folded PN modelling two identical resources;
b) The associated Markov chain.

$$\mu_{t_2}(M_x) = \begin{cases} 2\mu & \text{if} \quad m_2 \geq 2 \\ \mu & \text{if} \quad m_2 = 1 \\ 0 & \text{if} \quad m_2 = 0 \end{cases}$$

In this last case the Markov chain generated by the PN of Figure 17a) is reported in Figure 17c).

Example 5 - In the mutual exclusion problem of Figure 8, if the two resources C_1 and C_2 are identical, we can fold the two simmetric parts of the PN of Figure 8 in the PN of Figure 18a). The stochastic properties of the system are retained by assigning to transition t_1 a firing rate proportional to the number of tokens in p_1. The Markov chain associated to the PN of Figure 18a) is reported in Figure 18b). Note that folding the PN of Figure 18a) corresponds exactly to lumping the Markov chain of Figure 16 into the Markov chain of Figure 18b) (whenever lumpability conditions exist).

9.3. IMMEDIATE AND TIMED PN TRANSITIONS

Many authors [6,11,21] have recognized that the use of SPN for modelling real systems involves the presence of very brief or *fast* transitions, whose duration is short, or even negligible, with respect to the time scale of the problem. Different techniques have been proposed to tackle this problem.

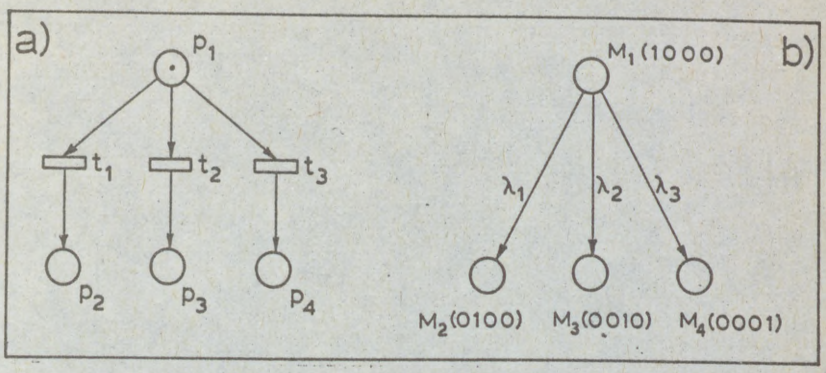


Figure 19 - a) SPN with timed transitions only;
b) The associated Markov chain.

The starting assumption in the GSPN model [6] is that it is desirable to associate a random time only to those transitions which are believed to have the largest impact on the system operation. Transitions are partitioned into two different classes: *immediate* transitions and *timed* transitions. Immediate transitions fire in zero time once they are enabled and have higher priority over timed transitions. Timed transitions fire after an exponentially distributed firing time. In the graphical representation of GSPN, immediate transitions are drawn as thin bars while timed transitions are drawn as thick bars.

Markings (states) enabling immediate transitions are passed through in zero time and are called *vanishing* states. Markings enabling only timed transitions are called *tangible*. Since the process spends zero time in the vanishing states, they do not contribute to the time behaviour of the system so that a procedure can be envisaged to eliminate them from the final Markov chain. With the partition of PN-transitions into a timed and an immediate class, we introduce a greater flexibility at the modelling level without increasing the dimensions of the final state space on which the set of equations (10) or (11) have to be computed.

Given a marking $M \in \mathcal{G}_R(M_1)$ of a GSPN, three different situations may arise:

Situation 1 (Figure 19)
Only timed transitions are enabled (Figure 19a) so that only tangible markings are generated (Figure 19b). The model, in this case, coincides with the HMSPN described in section 9.1.

Situation 2 (Figure 20)
Timed transitions are enabled simultaneously to one immediate transition (Figure 20a). Only the immediate transition is allowed to fire, generating the associate Markov chain of Figure 20b). However, marking M_2 is vanishing and can be eliminated from the chain producing the reduced Markov chain of Figure 20c), in which all the states are tangible.

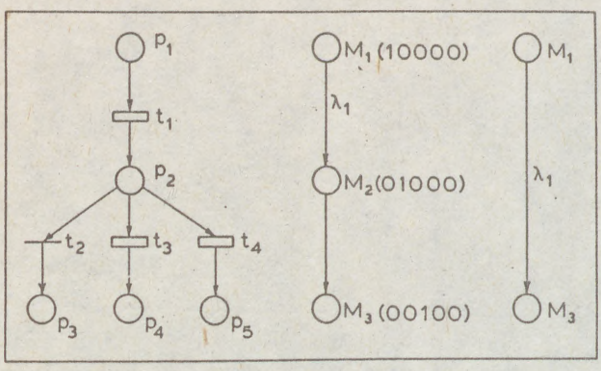


Figure 20 - a) SPN with one immediate transition;
b) The reachability graph;
c) The reduced Markov chain defined over tangible markings.

Situation 3 (Figure 21)
Several immediate transitions are enabled in a marking. In this case in order to define which is the transition that fires first, a probability mass function need to be specified: in the language of GSPN this construct is called a *random switch*, and the probability mass function the *switching distribution*. In Figure 21a) the immediate transition t_2 fires with probability v and the immediate transition t_3 with the complementary probability $1 - v$. The equivalent Markov chain is reported in Figure 21b). State M_2 is vanishing and can be eliminated incorporating the switching distribution into the rates leading to state M_2. Elimination of the vanishing state leads to the Markov chain of Figure 21c), which contains only tangible states.

An automatic algorithm can be implemented [6] which recognizes the three situations previously depicted and progressively eliminates vanishing states until a homogeneous Markov chain, defined over tangible states only, is obtained. In this way the reduction procedure becomes completely transparent to the analyst.

The problem of modelling a probabilistic decision, which does not consume time was also considered in [19], by introducing a different construct called *probabilistic arc*. For a comparison of probabilistic arcs with random swithches see [20].

In GSPN [6] only the steady state behaviour of the associated Markov chain is analysed. If the transient analysis is of interest, the use of immediate transitions does not allow to capture the true dynamics of the PN. In this case it is more appropriate to partition the PN-transition into *fast* transitions and *slow* transitions [11]. In this way the transition rate matrix Λ contains rates of very different orders of magnitude, so that the system of differential equations (10) becomes stiff [34]. The increase in the computational load

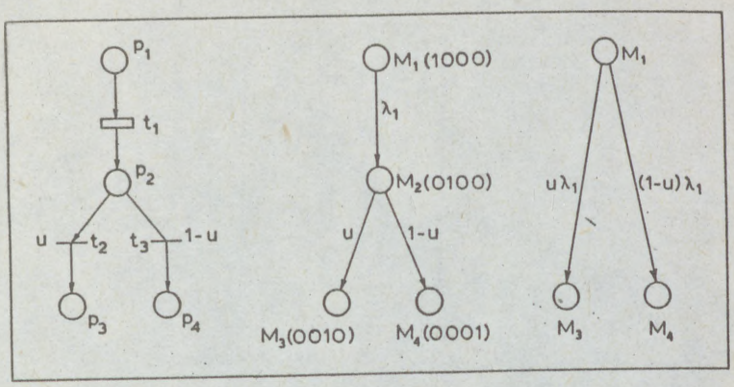


Figure 21 - a) The random switch;
b) The reachability graph;
c) The reduced Markov chain defined over tangible markings.

due to stiffness can be overcome by resorting to a decomposition technique [11,12]. This technique consists in decomposing the transition rate matrix Λ of the associated Markov chain in partitions, such that each partition contains rates of the same order of magnitude. An approximate solution to the original problem is obtained by solving each non-stiff partition in isolation [17]. This technique is incorporated in the package ESP [18].

Example 6 - In the folded PN of Figure 5 (the mutual exclusion problem with identical resources) transition t_2 has only the function of regulating the access to the shared resource C_s and thus can be modelled by an immediate transition (neglecting, in this case, the access time). The reachability set has 5 states (Figure 18a); markings M_2 and M_3 are vanishing since in these states the immediate transition t_2 is enabled. Eliminating the vanishing states by means of the previous rules leads to the reduced Markov chain of Figure 22 defined over tangible states only.

10. Computation of Measures of Reliability and Performance

A very important point of the time dependent representation of the system behaviour through SPN, is that they allow the user to define in a simple and natural way a large number of different measures related to the performance and reliability features of the system [7,20]. In order to exploit this peculiarity, the input language must be structured for providing a friendly environment for the specification of the output measures. In the sequel we refer in particular to the language of the ESP package [18].

The stochastic behaviour of a SPN is determined by calculating the occurrence probabilities over the states of the reachability set $\mathcal{R}(M_1)$. Therefore, the output measures are

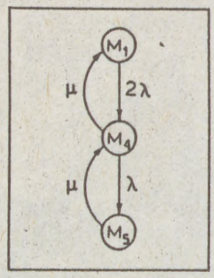


Figure 22 - Markov chain defined over the tangible states of the PN of Figure 5.

defined at the net level, and the numerical computation is carried out automatically by solving the associated equation (10) and by scanning the states in $\mathcal{R}(M_1)$.

Since some of the output measures depend on the integral of the probabilities rather than on the probabilities themselves [25], it is necessary to provide the package with the appropriate computation of the integral of the state probabilities. In the following discussion it is implicitly intended that time t ranges from 0 to ∞, so that all definitions apply to the transient as well as to the steady state solution.

10.1. PROBABILITY OF A GIVEN CONDITION ON THE SPN

By means of logical or algebraic functions of the number of tokens in the PN places, we can specify an output condition (e.g. no tokens in the failed place). We identify in $\mathcal{R}(M_1)$ the subset of places S for which the output condition is true. The output measure

$$Q_S(t) = Prob\ \{condition\ is\ true\ at\ time\ t\ \}$$

is given by:

$$Q_S(t) = \sum_{s \in S} q_s(t) \tag{12}$$

where $q_s(t)$ is the probability of being in state s at time t. For instance, if S is the set of operational states, $Q_S(t)$ in (12) is the usual definition of reliability (or availability).

A very useful case arises when we want to calculate the transient probability that the condition is satisfied for the first time. By using a standard device in the analysis of stochastic processes, we make the states $s \in S$ absorbing, and evaluate this quantity by stopping the process in S. An investigation of the application of SPN for computing the distribution of the completion time as a first marking problem is provided in [10].

10.2. TIME SPENT IN A MARKING

Let S be the subset of markings in which a particular condition is fulfilled. The expected time $\psi_S(t)$ spent in the markings $s \in S$ in the interval $0 - t$ is given by [8]:

$$\psi_S(t) = \sum_{s \in S} \int_0^t q_s(z)\, dz \tag{13}$$

Moreover, it is well known from the theory of Markov chains that as t approaches infinity the proportion of the time spent in states $s \in S$ equals the asymptotic probability:

$$Q_S(\infty) = \sum_{s \in S} q_s(\infty) \tag{14}$$

If S is the set of working states, $\psi_S(t)$ is the expected interval availability [25].

10.3. MEAN PASSAGE TIME

Given that $Q_S(t)$, as calculated in (12), is the probability of having entered subset S before t for the first time, the mean first passage time ϕ_S, has the usual expression:

$$\phi_S = \int_0^\infty [1 - Q_S(z)]\, dz \tag{15}$$

The above formula requires the transient analysis to be extended over long intervals. Of course, in this case, other well known direct techniques can be more effective [14].

10.4. DISTRIBUTION OF TOKENS IN A PLACE

Let p_i be a generic place of the PN. The cumulative distribution function (Cdf) of the number of tokens in p_i at time t is a staircase function in which the amplitude of the k-th step is obtained by summing up the probabilities of all the markings in $\mathcal{R}(M_1)$ containing k tokens ($k = 0, 1, 2, \ldots$) in p_i at time t. The density $f_i(k,t)$ is a mass function equal to the amplitude of the k-th step. The expected value of the number of tokens in place p_i at time t is:

$$E[m_i(t)] = \sum_{k=0}^{\infty} k\, f_i(k,t) \tag{16}$$

As an example, if place p_i represents identical units queueing up for a common resource the above quantities are the Cdf and the expected value of the number of units in the queue versus time. In reliability analysis a very interesting case arises when place p_i represents failed components. The above quantities provide the Cdf and the expected value of the number of failed components at time t.

10.5. EXPECTED NUMBER OF FIRINGS OF A PN-TRANSITION

Given an interval $(0,t)$ this quantity indicates how many times, on the average, an event modelled by a PN transition has occurred in that interval. Let t_k be a generic PN transition, and let S be the subset of $\mathcal{R}(M_1)$ which includes all the markings $s \in S$ enabling t_k. The expected number of firings of t_k in $(0,t)$ is given by:

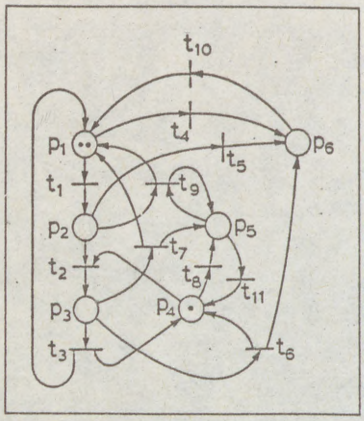


Figure 23 - System of Figure 18 with failures and repairs.

$$\eta_k(t) = \sum_{s \in S} \int_0^t q_s(z)\, \lambda_k(s)\, dz \tag{17}$$

where $\lambda_k(s)$ is the firing rate of t_k in marking s.
In steady state, the expected number of firings per unit time becomes:

$$\nu_k = \sum_{s \in S} q_s(\infty)\, \lambda_k(s) \tag{18}$$

where $q_s(\infty)$ is the steady state probability of state s. As an example, if transition t_k indicates failure (repair) of a component, $\eta_k(t)$ in (17) provides the mean number of failures (repairs) of that component in $(0, t)$.

11. Performance/Reliability Modelling through SPN

Performance-oriented reliability analysis has been the subject of an extensive literature in recent years [9,33,27,45]. We will show, by means of fully elaborated examples, that the SPN language, described in the previous sections, is very suitable to model this class of problems.

11.1. PARALLEL UNITS WITH SHARED RESOURCE

This situation has been depicted in Figure 8, and arises very often in distributed systems. With reference to Figure 8 in a multiprocessor system C_1 and C_2 are independent processors working locally on their private memories and C_s is a shared global memory which contains common data for the two processors. In a manufacturing system C_1 and C_2 are two working cells connected to the same transportation system or to the same load/unload device C_s.

Assuming C_1 and C_2 to be identical units, the SPN modelling the fault free system operation is reported in Figure 18. Taking into account the failure and repair of each unit the system operation is modelled by the SPN of Figure 23 [11].

TABLE I

Meaning of places and transitions in the SPN of Figure 23

p_1	Unit working independently	
p_2	Unit waiting for access to C_s	
p_3	Unit operating with C_s	
p_4	C_s free	
p_5	C_s failed	
p_6	Unit failed	
		firing rate
t_1	Unit requesting access to C_s	$1\ m_1$
t_2	Unit accessing C_s	10^4
t_3	Unit releasing C_s	5
t_4	Unit failure in local mode	$10^{-4}\ m_1$
t_5	Unit failure while waiting	$10^{-4}\ m_2$
t_6	Unit failure when working with C_s	10^{-4}
t_7	C_s failure while working	10^{-4}
t_8	C_s failure while free	10^{-4}
t_9	Return to local mode when C_s failed	10_4
t_{10}	Unit repair	10^{-2}
t_{11}	C_s repair	10^{-2}

Table I reports the meaning of the places and transitions of Figure 23, and the numerical values assigned to the firing rates associated to each PN transition. With the initial marking M_1 shown in Figure 10, the reachability set $\mathcal{R}(M_1)$ consists in 15 states whose token distribution is reported in Table II. By inspection of Tables I and II the following subsets of states can be recognized:

- States 1,2,5,6,11: fault-free operation of the system.

- States 3,7,13: normal operation of one unit when the other one is in a failed condition.

- States 4,8,12: two units operating and the shared resource failed.

- States 10,14: one unit operating while the other one and the shared resource failed.

- State 9: two units failed.

- State 15: two units and the shared resource failed.

TABLE II

Reachability set and token distribution of the SPN of Figure 23

State	Marking					
	m_1	m_2	m_3	m_4	m_5	m_6
1	2	0	0	1	0	0
2	1	1	0	1	0	0
3	1	0	0	1	0	1
4	2	0	0	0	1	0
5	0	2	0	1	0	0
6	1	0	1	0	0	0
7	0	1	0	1	0	1
8	1	1	0	0	1	0
9	0	0	0	1	0	2
10	1	0	0	0	1	1
11	0	1	1	0	0	0
12	0	2	0	0	1	0
13	0	0	1	0	0	1
14	0	1	0	0	1	1
15	0	0	0	0	1	2

Table III is the literal description of the reachability graph $\mathcal{G}_R(M_1)$ of the SPN; for each state of $\mathcal{R}(M_1)$ on the first column, the enabled transitions and the immediately reachable states (in parentheses) are reported. Substituting the numerical values of the firing rates reported in Table I to the transition labels of Table III the transition rate matrix Λ of the associated Markov chain can be automatically generated.

By considering the access time to $\mathbf{C}_s$ as negligible with respect to the time constant of the system, t_2 and t_9 can be interpreted as immediate transitions. With this assumption it is seen from Table III that the states { 2,5,7,8,12,14 } become vanishing since in these states one of the immediate transitions is enabled. By reducing the state space with the rules of section 9.3, the final Markov chain is defined over a state space containing 9 tangible states.

An interesting performance/reliability measure for this system is the number of units doing useful work at time t, where by useful work we mean the work performed by each unit when operating independently. This measure takes into account the reduction in the

system performance due to different effects: the congestion delays due to the sharing of the common resource, the transfer of data or pieces from each unit to C_s and the failure and repair cycles. By using the definitions of the previous section and looking at Table I, it is seen that this measure coincides with the expected number of tokens in place p_1 and thus can be easily defined at the PN level and computed by means of Equation (16).

TABLE III

Literal description of the Reachability Graph $\mathcal{G}_R(M_1)$

State	Enabled transition and immediately reachable state									
1	1	(2)	4	(3)	8	(4)				
2	1	(5)	2	(6)	4	(7)	5	(3)	8	(8)
3	1	(7)	4	(9)	8	(10)	10	(1)		
4	1	(8)	4	(10)	11	(1)				
5	2	(11)	5	(7)	8	(12)				
6	1	(11)	3	(1)	4	(13)	6	(3)	7	(4)
7	2	(13)	5	(9)	8	(14)	10	(2)		
8	1	(12)	4	(14)	5	(10)	9	(4)	11	(2)
9	8	(15)	10	(3)						
10	1	(14)	4	(15)	10	(4)	11	(3)		
11	3	(2)	5	(13)	6	(7)	7	(8)		
12	5	(14)	9	(8)	11	(5)				
13	3	(3)	6	(9)	7	(10)	10	(6)		
14	5	(15)	9	(10)	10	(8)	11	(7)		
15	10	(10)	11	(9)						

11.2. PARALLEL SYSTEM WITH FINITE INPUT BUFFER

The block diagram of the system is shown in Figure 24. It consists in u identical units $U_1, U_2, \ldots, U_u$ and in an input buffer with b positions $B_1, B_2, \ldots, B_b$ [33].

The GSPN model of the fault free system operation is shown in Figure 25 [7]; the sum of tokens in p_1 and p_2 is equal to b (number of buffer positions; see also Section 4.3), whereas the sum of tokens in p_3 and p_4 is equal to u (number of parallel units). In other words, $\{p_1, p_2\}$ and $\{p_3, p_4\}$ form place-invariants.

The firing rate associated to t_1 is the task arrival rate λ, while the firing rate associated to t_3 is the service rate proportional to the number of active units $m_4\,\mu$, being μ the service rate of a single unit and m_4 the number of tokens in p_4. t_2 is an immediate transition (we neglect the transfer time from the buffer to the service station).

When failures and repairs are considered, the GSPN model becomes as in Figure 26. Heavy lines represent the fault-free operation, light lines failures and dotted lines repairs.

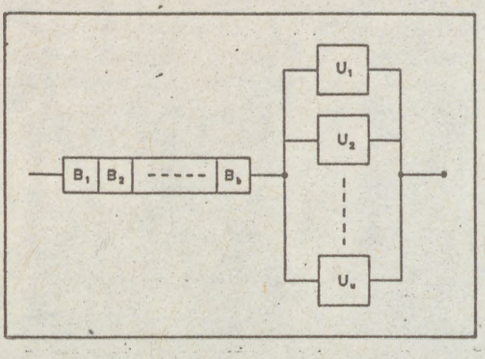


Figure 24 - Block diagram of a parallel system with finite buffer.

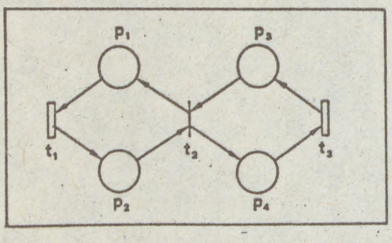


Figure 25 - Fault-free SPN model of the system of Figure 24.

Let us first focus our attention on the failure transitions; with reference to Figure 26 the following hypotheses have been considered:

- Buffer stages fail one at the time, either when free (t_4), or when occupied (t_5), with possibly different failure rates. t_6 and t_7 form a random switch modelling the fact that with probability v_B a buffer stage failure is recovered (the buffer continues to be operational with a storing capacity reduced by one stage), and with probability $1 - v_B$ the failure is not recovered and the buffer locks (inhibitor arc from p_7 to t_2).

- The units U_i $(i = 1, 2, \ldots, u)$ fail either when idle (t_8) or when active (t_9), with possibly different failure rates. The failure of an idle unit is recovered with probability one, while the failure of an active unit is recovered with probability v_U (random switch t_{10} t_{11}). A task is lost only when an active unit fails.

. By slightly modifying the GSPN of Figure 26, different design alternatives or recovery strategies could be accommodated. When repair is considered, t_{12} and t_{13} refer to buffer

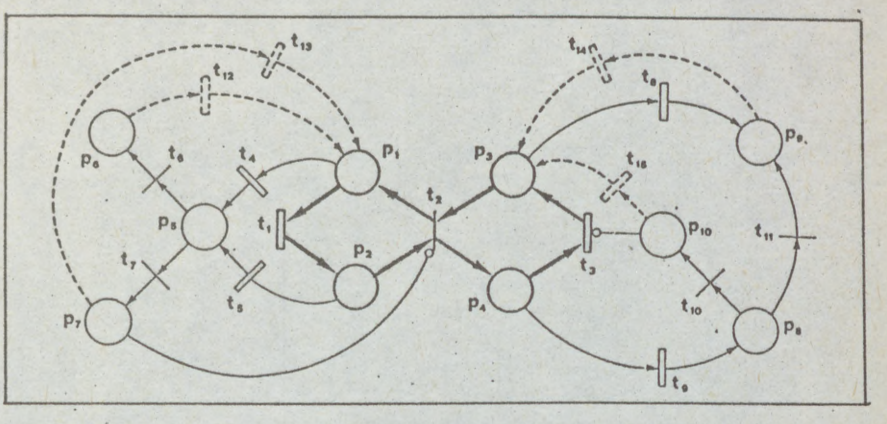


Figure 26 - SPN model of the system of Figure 24 with failures and repairs.

stage repair, and t_{14} and t_{15} to processor repair; the considered model allows us to allocate different repair rates for recoverable and unrecoverable failures.

The meaning of places and transitions in Figure 26 is summarized in Table IV, where the expressions of the firing rates for the timed transitions, and of the switching probabilities for the immediate transitions, are also given.

The initial marking M_1 consists in b tokens in place p_1 and u tokens in p_3. As measures characterizing the system performance and reliability, we define the following.

. *Mean fraction of arrived tasks processed in 0-t.*

The mean number of processed tasks in $0-t$ is given by the mean number of firings of t_3 (Equation 17). The mean number of arrived tasks in $0-t$ is simply $\lambda \cdot t$, having assumed a Poisson arrival process with rate λ. Thus the performance/reliability index $Y(t)$, representing the mean fraction of arrived tasks processed in $0-t$ is calculated as:

$$Y(t) = \frac{\eta_3(t)}{\lambda t} \tag{19}$$

. *Mean number of failures (repairs) in 0-t.*

This quantity is given by $[\eta_4(t)+\eta_5(t)]$ (Equation 17) for buffer stage failure ($[\eta_{12}(t)+\eta_{13}(t)]$ for buffer stage repair), and by $[\eta_8(t)+\eta_9(t)]$ for unit failure ($[\eta_{14}(t)+\eta_{15}(t)]$ for unit repair).

. *Cdf and mean number of active, idle, failed, units or buffer stages.*
These quantities are obtained by applying the procedure of paragraph 10.4 to place p_4 for active units, to place p_3 for idle units, and to places $[p_9 + p_{10}]$ for failed units. Similarly, place p_1 indicates free buffer stages, p_2 filled buffer stages, and $[p_6 + p_7]$ failed buffer stages.

TABLE IV

Meaning of places and transitions in the SPN of Figure 26

p_1	Free buffer stage	
p_2	Occupied buffer stage	
p_3	Idle unit	
p_4	Active unit	
p_5	Failed buffer stage	
p_6	Recovered buffer stage failure	
p_7	Unrecovered buffer stage failure	
p_8	Failed active unit	
p_9	Recovered unit failure	
p_{10}	Unrecovered unit failure	
		firing rate
t_1	Buffer stage becomes occupied	λ
t_2	Transfer from buffer to unit	*immed.*
t_3	Unit ends a task	$m_4\,\mu$
t_4	Free buffer stage fails	$m_1\,\gamma_4$
t_5	Occupied buffer stage fails	$m_2\,\gamma_5$
t_6	Buffer stage failure is recovered	v_B
t_7	Buffer stage failure is not recovered	$(1 - v_B)$
t_8	Idle unit fails	$m_3\,\gamma_8$
t_9	Active unit fails	$m_4\,\gamma_9$
t_{10}	Unit failure is not recovered	$(1 - v_U)$
t_{11}	Unit failure is recovered	v_U
t_{12}	Repair of recovered buffer stage	ρ_{12}
t_{13}	Repair of unrecovered buffer stage	ρ_{13}
t_{14}	Repair of recovered unit	ρ_{14}
t_{15}	Repair of unrecovered unit	ρ_{15}

A numerical example has been run with $u = 2$ and $b = 2$. The reachability set, in this case, comprises 88 tangible states and 84 vanishing states. With reference to Table IV, we have assigned to the parameters the following numerical values (being $w = \lambda/\mu$ the load factor of the system):

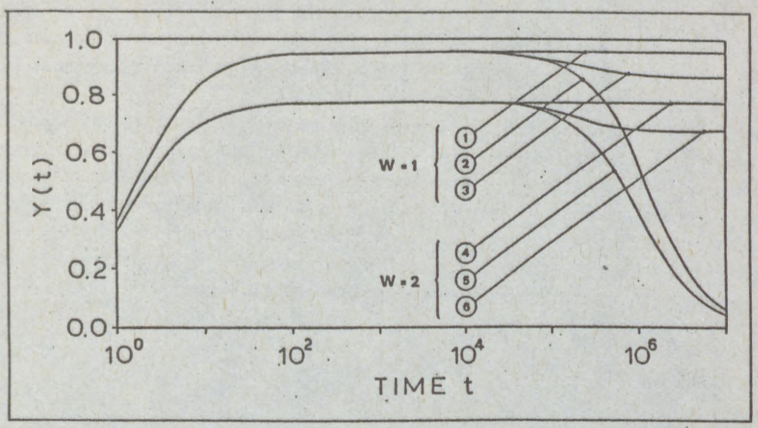


Figure 27 - Mean fraction of arrived tasks processed in $0 - t$ versus time.

$$
\begin{aligned}
\mu &= 1 \\
\lambda &= w\,\mu \\
\gamma_4 &= \gamma_5 = \gamma_8 = \gamma_9 = \gamma = 1.0\,10^{-6} \\
\rho_{12} &= \rho_{13} = \rho_{14} = \rho_{15} = 10\,\gamma \\
v_B &= v_U = 0.9
\end{aligned}
$$

Two cases have been examined with different load factors: $w = 1$ and $w = 2$. The last case represents the ideal load factor since the arrival rate is twice as large as the service rate, but there are two parallel service units. Numerical results have been obtained using the program ESP [18] and resorting to a decomposition technique due to the high spread in the firing rate values. Figure 27 shows $Y(t)$ (Equation 19) as a function of t for the two chosen values of w, and in three different conditions, namely: fault-free operation (curves 1 and 4); with failures (curves 2 and 5); with failures and repairs (curves 3 and 6). Figure 27 shows how the system performance (throughput) is degraded when considering failures and failures/repairs, and can be of valuable support at the design level.

12. Simulative Analysis of SPN

In the previous Sections the SPN was used as a language for generating an associated Markov chain whose transient and ergodic behaviour is obtained by solving Equations (10) and (11) respectively. However, a simulative approach is also possible.

The simulative approach is very simple from a logical point of view, and is easily implementable in a computer program, so that SPN can also be considered as a possible general simulative language. Due to these characteristics, it is conceivable to construct general SPN solvers [20] where both the analytic approach (when feasible and convenient) and the simulative approach are present.

The core of the simulator is that in each marking we need to choose the PN transition which actually fires, among those enabled. This choice is done by generating a random sample from the distribution function associated to each transition and selecting the transition with the minimum firing sample. The simulator clock is updated with the minimum sample and we move to the next marking where a new choice procedure is initiated. The basic algorithm for the generation of a timed execution sequence $\mathcal{T}_E$ (section 8) can be outlined as follows, when the firing times are exponentially distributed:

```
begin
marking ← initial marking
clock = 0
repeat
   for j := 1 to n_t do
      begin
      if { t_k is enabled } then
         generate a random sample θ_k
      end
   find minimum θ_k
   generate new marking
   clock=clock+min(θ_k)
until { terminating condition is fulfilled }
end
```

The termination criteria are driven by the type of simulation: transient simulation or ergodic simulation [19]. In the transient simulation the user defines exit places or absorbing places; the simulation trial is stopped once a token reaches an exit place. Statistics are gathered by generating random timed execution sequences $\mathcal{T}_E$ through the PN [22].

The ergodic simulation is a regenerative-type simulation [30], in which a return to the initial marking constitutes a regeneration point in the simulation. A trial is defined as the random timed execution sequence $\mathcal{T}_E$ starting and ending with the initial marking.

All the measures defined in Section 10 can be estimated as a result of the simulation approach. The definition of these measures in both transient and ergodic simulation is usually straightforward, and is outlined in [20]. Confidence intervals can be also calculated as a function of the number of trials [19].

The very important fact about the simulative approach, to be noted here, is that in each trial we only generate a single timed execution sequence $\mathcal{T}_E$, so that we do not need to generate and store all the reachable markings at the same time. Moreover, the extension of the simulative approach to the case in which the random variables associated to the PN transitions are generally distributed is, in principle, quite simple. In fact, once the execution policy is specified (i.e. the way in which the SPN keeps trace of the past history; see Section 8), the basic simulation algorithm must be modified by attaching a clock to

each PN-transitions. Each time a move is selected, the clocks are updated by recovering the elapsed time as specified by the execution policy, and the following selection is performed by comparing the values of all these clocks. This extension [21,26] is not further considered in the present notes.

13. Conclusion

These lecture notes were intended as introductive material to the use of Petri nets as a general language for the modelling and analysis of the behaviour of complex systems versus time. In the first part, the aim was to show how the semantics of classical PN is suitable to model various kinds of logical as well as physical interactions among components in a system (interactions that are not easily reprensentable in other modelling frameworks).

The second part was more specifically devoted to define the Stochastic PN extension and to present examples taken from the reliability area. Only the case where the stochastic process associated to the SPN is a homogeneous Markov chain has been considered in details. This case arises when the firing times assigned to the PN transitions are exponentially distributed.

From the discussion contained in these lecture notes we can summarize some advantages and disadvantages of the SPN as a modelling tool. The main advantages include: the graphic nature, the conciseness in comparison with state graphs, the possibility of implementing analysis techniques. The graphic nature facilitates the use by non skilled users and allows to implement very friendly graphic editors for the specification of the input net. We finally stress that the use of SPN requires only the specification of the topology of the starting PN, the specification of the firing rates (or of the distribution functions in the general case) associated to the transitions and the specification of the output measures to be computed following the indications provided in Section 10. All the subsequent steps, which consist in:

- the generation of the reachability graph $\mathcal{G}_R(M_1)$;
- the generation of the associated Markov chain;
- the transient and ergodic solution of the Markov chain;
- the evaluation of the relevant process measures;

can be executed in a completely automated way by a computer program, thus making transparent to the user the associated mathematics.

The main disadvantages of SPN arise from the size of the net obtained in modelling very complex distributed systems. In this case the model is difficult to validate at the net level, and the number of reachable markings tends to explode, making analytically intractable the associated Markov chain. It should be recognized, however, that this drawback is common to almost all general purpose modelling techniques.

References

[1] *International Workshop Petri Nets and Performance Models*, IEEE Computer Society Press No. 796, Madison, 1987.

[2] *International Workshop Timed Petri Nets*, IEEE Computer Society Press No. 674, Torino (Italy), 1985.

[3] T. Agerwala. Putting Petri nets to work. *Computer*, 85–94, December 1979.

[4] M. Ajmone Marsan, G. Balbo, A. Bobbio, G. Chiola, G. Conte, and A. Cumani. The effect of execution policies on the semantics and analysis of stochastic Petri nets. *To be published on: IEEE Transactions on Software Engineering*, 1989.

[5] M. Ajmone Marsan, G. Balbo, A. Bobbio, G. Chiola, G. Conte, and A. Cumani. On Petri nets with stochastic timing. In *Proceedings International Workshop on Timed Petri Nets*, pages 80–87, IEEE Computer Society Press no. 674, Torino (Italy), 1985.

[6] M. Ajmone Marsan, G. Balbo, and G. Conte. A class of generalized stochastic Petri nets for the performance evaluation of multiprocessor systems. *ACM Transactions on Computer Systems*, 2:93–122, 1984.

[7] M. Ajmone Marsan, A. Bobbio, G. Conte, and A. Cumani. Performance analysis of degradable multiprocessor systems using generalized stochastic Petri nets. *IEEE Computer Society Newsletters*, 6, SI-1:47–54, 1984.

[8] R.E. Barlow and F. Proschan. *Statistical Theory of Reliability and Life Testing*. Holt, Rinehart and Winston, New York, 1975.

[9] M.D. Beaudry. Performance-related reliability measures for computing systems. *IEEE Transactions on Computers*, C-27:540–547, 1978.

[10] A. Bobbio. Petri nets generating Markov reward models for performance reliability analysis of degradable systems. In *Proceedings of the 4-th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*, pages 413–431, Palma de Mallorca, 1988.

[11] A. Bobbio, A.Cumani, and R. Del Bello. Reduced markovian representation of stochastic Petri net models. *Systems Science*, 10:5–23, 1984.

[12] A. Bobbio and K.S. Trivedi. An aggregation technique for the transient analysis of stiff Markov chains. *IEEE Transactions on Computers*, C-35:803–814, 1986.

[13] G.W. Brams. *Réseaux de Petri: Théorie et pratique*. Masson, 1983. (in French).

[14] J.A. Buzacott. Markov approach to finding failure times of repairable systems. *IEEE Transactions on Reliability*, R-19:128–134, 1970.

[15] W.M. Chow, E.A. McNair, and C.H. Sauer. Analysis of manufacturing systems by Research Queueing Package. *IBM Journal of Research and Development*, 29:330–341, 1985.

[16] G. Ciardo. Toward a definition of modeling power for stochastic Petri net models. In *Proceedings International Workshop on Petri Nets and Performance Models*, pages 54–62, IEEE Computer Society Press no. 796, Madison, 1987.

[17] P.J. Courtois. *Decomposability: Queueing and Computer System Applications.* Academic Press, New York, 1977.

[18] A. Cumani. Esp - A package for the evaluation of stochastic Petri nets with phase-type distributed transition times. In *Proceedings International Workshop Timed Petri Nets*, IEEE Computer Society Press no. 674, Torino (Italy), 1985.

[19] J. Bechta Dugan. *Extended stochastic Petri nets: applications and analysis.* Technical Report, Phd Thesis, Department of Computer Science, Duke University, 1984.

[20] J. Bechta Dugan, A. Bobbio, G. Ciardo, and K. Trivedi. The design of a unified package for the solution of stochastic Petri net models. In *Proceedings International Workshop on Timed Petri Nets*, pages 6–13, IEEE Comp Soc Press no. 674, Torino (Italy), 1985.

[21] J. Bechta Dugan, K. Trivedi, R. Geist, and V.F. Nicola. Extended stochastic Petri nets: applications and analysis. In *Proceedings PERFORMANCE '84*, Paris, 1984.

[22] G.S. Fishman. *Concepts and methods in discrete event digital simulation.* Wiley, New York, 1973.

[23] G. Florin and S. Natkin. Les reseaux de Petri stochastiques. *Technique et Science Informatique*, 4:143–160, 1985.

[24] H.J. Genrich and K. Lautenbach. System modelling with high level Petri nets. *Theoretical Computer Science*, 13:109–136, 1981.

[25] A. Goyal, S. Lavenberg, and K.S. Trivedi. Probabilistic modeling of computer system availability. *Annals of Operations Research*, 8:285–306, 1987.

[26] P.J. Haas and G.S. Shedler. Regenerative stochastic Petri nets. *Performance Evaluation*, 6:189–204, 1986.

[27] B.R. Iyer, L. Donatiello, and P. Heidelberger. Analysis of performability for stochastic models of fault-tolerant systems. *IEEE Transactions on Computers*, C-35:902–907, 1986.

[28] K. Jensen. Coloured Petri nets and the invariant method. *Theoretical Computer Science*, 14:317–336, 1981.

[29] L. Kleinrock. *Queuing systems, Volume 1: Theory.* Wiley Interscience, New York, 1975.

[30] A.J. Lemoine M.A. Crane. An introduction to the regenerative method for simulation analysis. In A.V. Balakrishnan and M. Thorna, editors, *Lecture Notes in Control and Information Sciences*, Springer-Verlag, 1977.

[31] J. Martinez and M. Silva. A simple fast algorithm to obtain all invariants of a generalized Petri net. In *Proceedings 2-nd European Workshop on Application and Theory of Petri Nets*, Springer-Verlag, 1981.

[32] P.M. Merlin and D.J. Faber. Recoverability of communication protocols - Implication of a theoretical study. *IEEE Transactions on Communication*, COM-24:1036–1043, 1976.

[33] J.F. Meyer. Closed form solution of performability. *IEEE Transactions on Computers*, C-31:648–657, 1982.

[34] W.L. Miranker. *Numerical Methods for Stiff Equations.* Reidel, Dordrecht, 1981.

[35] M.K. Molloy. *On the integration of delay and throughput measures in distributed processing models.* Technical Report, Phd Thesis, UCLA, 1981.

[36] S. Natkin. *Les reseaux de Petri stochastiques et leur application a l'evaluation des systemes informatiques.* Technical Report, These de Docteur Ingegneur, CNAM, Paris, 1980.

[37] J.L. Peterson. *Petri net theory and the modeling of systems.* Prentice Hall, Englewood Cliffs, 1981.

[38] J.L. Peterson. Petri nets. *Computing Surveys*, 9:223–252, 1977.

[39] C.A. Petri. *Kommunikation mit automaten.* Technical Report, Doctoral Thesis, University of Bonn, 1962. (Available in English as: *Communication with automata*, Technical Report RADC-TR-65-377, Rome Air Development Center, Griffis NY, 1966).

[40] C.V. Ramamoorthy and G.S. Ho. Performance evaluation of asynchronous concurrent systems using Petri nets. *IEEE Transactions on Software Engineering*, SE-6:440–449, 1980.

[41] A. Reibman and K.S. Trivedi. Numerical transient analysis of Markov models. *Computers and Operations Research*, 15:19–36, 1988.

[42] W. Reisig. *Petri nets - An introduction.* Springer-Verlag, 1982.

[43] J. Sifakis. Use of Petri nets for performance evaluation. In H. Beilner and E. Gelenbe, editors, *Measuring, modelling and evaluating computer systems*, pages 75–93, North Holland, 1977.

[44] M. Silva. *Las redes de Petri en la Automatica y la Informatica.* AC, Madrid, 1985.

[45] R. Smith, K. Trivedi, and A.V. Ramesh. Performability analysis: measures, an algorithm and a case study. *IEEE Transactions on Computers*, C-37:406–417, 1988.

[46] W. Whitt. Blocking when service is required from several facilities simultaneously. *AT&T Technical Journal*, 64:1807–1856, 1985.

[47] W.M. Zuberek. Timed Petri nets and preliminary performance evaluation. In *Proceedings 7-th Annual Symposium on Computer Architecture*, pages 88–96, 1980.

APPLICATION OF ADVANCED FAULT TREE ANALYSIS TO INDUSTRIAL RELIABILITY PROJECTS

David J. BURNS, WS Atkins Engineering Sciences
United Kingdom

Application of Advanced Fault Tree Analysis to Industrial Reliability Projects

D J Burns
WS Atkins Engineering Sciences
Woodcote Grove, Ashley Road
Epsom, Surrey KT18 5BW, UK

ABSTRACT

The application of Fault Tree Analysis (FTA) to modern industrial projects can assist the assessment of safety, availability and Life Cycle Costs (LCC). The roles of the operator, the equipment vendor and the reliability engineer in arriving at an acceptable target for safety, reliability and costs is discussed, with emphasis on the use of FTA. A means of approaching set targets of safety, availability and cost is described, using an integrated software package for fault tree construction and analysis: SUPER-NET.

1. INTRODUCTION

Among the numerous criteria which must be met by modern industrial plants[1], two are discussed in this paper where the use of Fault Tree Analysis (FTA) can be of great assistance..

The first criterion is that of safety and, specifically, the need to demonstrate that hazardous events can be safely contained by reliable contingency operations and systems. The risk of damaging consequences to human beings, plant or environment must be shown to be acceptably low. Thus Section 2 presents a schematic model for the main steps employed in a Probabilistic Safety Assessment (PSA), indicating where FTA is applied.

The second criterion is that of plant operational availability, and specifically, the need to demonstrate that the availability targets can be met at, or near to, the optimal cost for the life time of the plant. Section 3, therefore, presents a further schematic model where the emphasis is on availability of plant and sub-systems, and on the interests of both operator and equipment vendors in demonstrating that the availability criteria will be met.

The role of availability in the Life Cycle Cost (LCC) of a plant is discussed in Section 4.

Section 5 comprises an overview of an integrated suite of programs (SUPER-NET) for FTA and LCC analyses which has been developed with the above needs for safety and protection of the investment in mind.

2. PROBABILISTIC SAFETY ASSESSMENT[2,3,4]

The essence of this type of study is to demonstrate that a plant is safe by assessing the level of risk associated with all identifiable major hazard events. The risk is generally stated as an estimated frequency of occurrence for a certain level of damage. The level of damage lies within the domain of consequence analysis, and will not be addressed here. However, the estimation of frequency of occurrence is one of the uses of FTA.

Figure 1 shows the main building blocks of a PSA. After ascertaining the workings of the plant, the accident initiating events are identified by various techniques such as Hazard and Operability Study (HAZOP), Failure Modes Effects and Criticality Analysis (FMECA) and surveys of case histories.

For each identified initiating event, an event tree is constructed whereby the worst possible accident scenarios are postulated. At each branch in the event tree, an event is defined which can aggravate the scenario if it occurs. Accident scenarios leading to Major Catastrophes are said to be initiated by Major Hazard Events. These are then quantified on two counts: firstly that their damage effect is calculated by physical models, and secondly that their frequency is estimated. This is often achieved by FTA, when the event is broken down into possible precursors, the estimated frequencies of which are combined using Boolean logic.

It has been noted that the event tree contains postulated aggravating events, whose probability of occurrence needs to be calculated in order to arrive at a final frequency estimation for the catastrophic event. Again FTA is an ideal means of arriving at branch probabilities.

The event tree analysis is carried out for several initiating events, and the frequencies of all like catastrophic events are summed from all initiating event considered in order to make comparison with acceptance criteria. Plants which do not meet the criteria will need to have some redesign, if at the design stage, or, if operational, some back-fitting.

3. PLANT AVAILABILITY ASSESSMENT[4,5]

Availability analyses of plant are often carried out as a function of time by simulation techniques. However, mean unavailability over a period of time can be estimated using FTA, and this is useful for vendors wishing to demonstrate the total availability of their systems or to optimize redundancy in equipment or spares holding.

Figure 2 shows the scheme for applying FTA to availability modelling. Starting with the plant model, more than one operational mode may be possible, the availability target for each operational mode being different. For each operational mode, a fault tree top event will be definable reflecting the frequency of failure of the plant. A fault tree can then be drawn up to indicate the possible causes of total plant failure which, when provided with failure and repair data for all basic system failure events will constitute the Integrated Plant Unavailability Model.

Each system failure is then made the top event of a separate fault tree, and a break-down of system failures into component failures carried out. As availability targets can be determined for each system, in order to meet the total plant availability target, it is possible to present vendors with availability targets for their equipment. In some cases the initial target established by the operator cannot be met by the vendor without extra cost, and negotiations may result in a compromise being reached. The FTA is very useful here in demonstrating the sensitivity of the total plant availability to each system's performance. So not meeting the original system availability target set by the operator could result in

a) resetting the target for the system availability
b) redesigning the system to meet the original target
c) redesigning the plant to meet the availability target

Resetting the plant availability target is possible, but unlikely. The above procedure would be repeated for each operational mode.

4. AVAILABILITY AND LIFE CYCLE COST (LCC)[4,5]

In additional to demonstration of a particular vendor's system availability, the operator will wish to calculate the total cost of procuring equipment, running and maintaining the plant, and of production loss when the plant stands idle.

System designers, reliability engineers and procurement staff should work together to arrive at the cost-optimized availability goal, taking into account initial equipment costs, levels of redundancy, maintenance costs, and cash flow.

The relationship between the parameters involved in LCC considerations is shown in Figure 3.

Availability of operation can, in theory, be increased more and more by investing in more and better equipment. Conversely, at a low level of investment cost, more operational costs are incurred due to plant breaking down. As investment increases, so the need for maintenance (operation costs) decreases. Thus the total cost (LCC) passes through a minimum. The reliability engineer, as coordinator between design and procurement, can assist greatly in getting the availability target near to the minimum LCC.

5. COMPUTERISED FAULT TREE AND LCC MODELLING[2,3,4,5]

The assessment of availability and its application to targets of safety, plant operability and LCC may be carried out quickly and effectively using the SUPER-NET package. Developed by ABB Atom in Sweden, this consists of the following units (Figure 4):

SUPER-TREE	for screen-orientated fault tree handling
CUTSET	for fault tree analysis
SENS	for importance and sensitivity analysis
FRANTIC	for time-dependent reliability analysis
SAMPLE	for statistical uncertainly analysis
COST	for Life Cycle Cost analysis

5.1 *SUPER-TREE*

This is a semi-automatic fault tree handling program which allows the fault tree to be built up interactively on the screen of a PC or Minicomputer. The tree structure is left-adjusted to enable an automatic assignment of gate addresses (Figure 5). Whole sections of the tree can be copied and relabelled automatically and checks are in place for errors in the tree structure. Drawing and restructuring of the tree can be carried out at two levels of detail, while a third level is reserved for details of basic event data including failure rate, repair time and cost and replacement cost. The data can be transferred automatically to the basic event in the fault tree by an event code system, or manually as required. The various levels are illustrated in Figure 6.

5.2 *CUTSET*

Top events of fault trees represent either the frequency of some event, such as total plant failure, or the failure on demand of a system or piece of equipment. Whichever type is under analysis, the combination of events which bring about the top event are called cut-sets, and the numerical analysis of the, often many, combinations of events may be carried out using a Boolean reduction by the CUTSET program. The cut-sets are presented in order of magnitude, and are easily identified by the user-specified event coding system. The total unavailability of frequency of failure for the top event is presented as the sum of the individual cut-set values.

5.3 *SENS*

The cut-sets, as calculated according to the preceding section, are summed to give a first-moment estimation of the total unavailability or failure frequency. This approach assumes no dependence between the cut-sets and is therefore an approximation, which is satisfactory providing no significant level of interaction between the failure modes, such as common causes, is applicable. If such an interaction is considered to be valid, more precise results may be generated using the SENS program. This performs sensitivity analysis and lists importance rankings on the results from the CUTSET analysis.

For the base case, Fussel-Veseley importance measures are generated for all the basic events in the cut-set list. This is then repeated by changing failure data for individual components, for groups of components, or classes of components, in order to perform a sensitivity analysis. Results are presented in bar chart and graphical form.

5.4 *FRANTIC*

The availability of stand-by systems varies with time since, in additional to the system components' failure probabilities, test intervals, and repairs of revealed failures, will contribute further to the picture. Thus FRANTIC creates an unavailability function for the system under analysis, based on the cut-set list generated by the CUTSET program. This function is compounded by individual component data from SUPER-TREE such as failure frequency, repair times and test intervals.

By providing lists and graphs of unavailability as a function of time, this program is useful for planning and evaluating the testing and maintenance of system components.

FRANTIC was originally developed by the US Nuclear Regulatory Commission.

5.5 *SAMPLE*

While the above programs all work from point values of failure probabilities, it is important to know the uncertainty distribution for key events. The unavailability function as generated from CUTSET by FRANTIC for the top event, is compounded from SUPER-TREE by details of distribution parameters for component failure rates and repair times, using the SAMPLE program. This program uses Monte-Carlo simulation to compute the uncertainty distribution for the top event.

SAMPLE was originally developed by the US Nuclear Regulatory Commission.

5.6 *COST*

The LCC of a plant comprises two basic components: the initial costs and the recurring costs. The COST program covers all aspects of these costs, some of which are specified by the user (interest rates, foreign exchange rates, etc.) some of which originate from SUPER-TREE (e.g. initial component costs, scheduled maintenance requirements) and some of which are generated by CUTSET (unavailability of the plant leading to production losses, corrective maintenance costs etc.).

A sensitivity analysis facility allows the LCC's dependence on key parameters to be assessed, in helping to keep costs at an optimum level with respect to availability targets.

6. APPLICATIONS

The analysis and programs described in this paper are applicable to any plant or system, large or small. The advantages to be gained in analysing large systems include the handling of the following tasks:

- drawing and data-setting of the first complete set of fault trees
- updating of the fault trees as the analysis proceeds and changes are introduced.
- co-ordination of the work of several analysts contributing to the whole study.

FTA, as part of a safety assessment, an availability assessment or a LCC analysis finds applications in many branches of modern technology.

Some examples are:

Power generation
Power transmission and distribution
Telecommunications
Aerospace
Transport
Chemical process industries
Oil and gas production transmission and distribution
Marine systems

Retaining models of fault trees and LCC on file throughout the installation's operational life provides a powerful management tool in assisting with decisions involving design, equipment, organisational or monetary fluctuations.

REFERENCES

1. Watson I A, "Safety and Reliability Procedures in Various Industries". Safety and Reliability Directorate UKAEA, SRS/GR/76, January 1989.

2. Hirschberg S, and Knochenhauer M. "SUPER-NET, a Multi-purpose Tool for Reliability and Risk Assessment". International Post-SMIRT 10 Seminar. "The Role and Use of PCs in Probabilistic Safety assessment and Decision Making". Beverley Hills, California, August 21-22, 1989.

3. Björe S, Hirschberg S, and Knochenhauer M . "A Unified Approach to Reliability Analysis". Society of Reliability Engineers Symposium, Vasteras, Sweden, October 10-12, 1988.

4. Hirschberg S et al. "A Comparative Uncertainty and Sensitivity of an Accident Sequence" ibid.

5. Knochenhauer M, Olsson L, and Alm S. "Verification of Availability Guarantees in HVDC Projects: Estimation and Optimisation of the Impact from Corrective and Preventive Maintenance". Reliability Achievement: The Commercial Incentive. SRE-Symposium, Stavanger, Norway, October 9-11, 1989.

ACKNOWLEDGEMENT

The author would like to thank ABB Atom, Västerås, Sweden, for permission to publish this paper.

USE OF FAULT TREE ANALYSIS IN PROBABILISTIC SAFETY ASSESSMENT

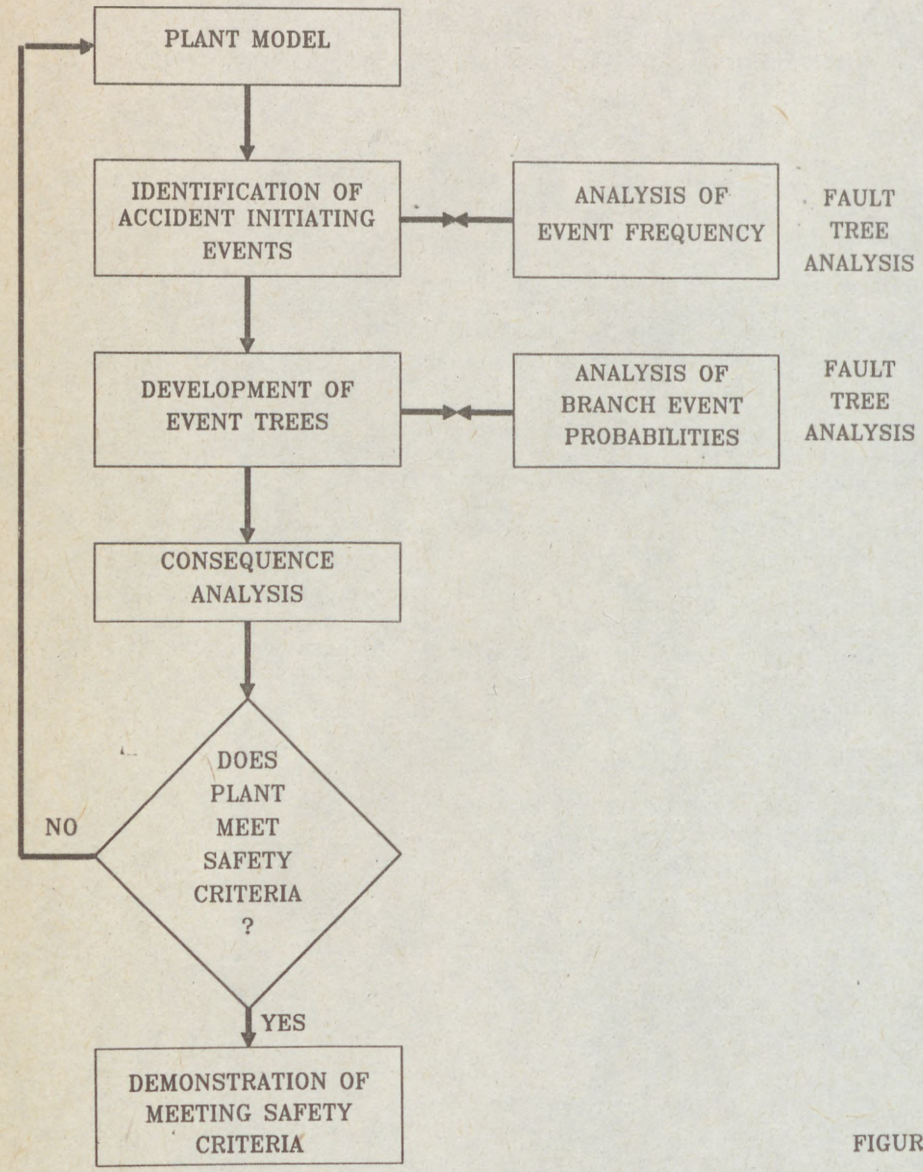


FIGURE 1

USE OF FAULT TREE ANALYSIS IN AVAILABILITY TARGET CALCULATIONS

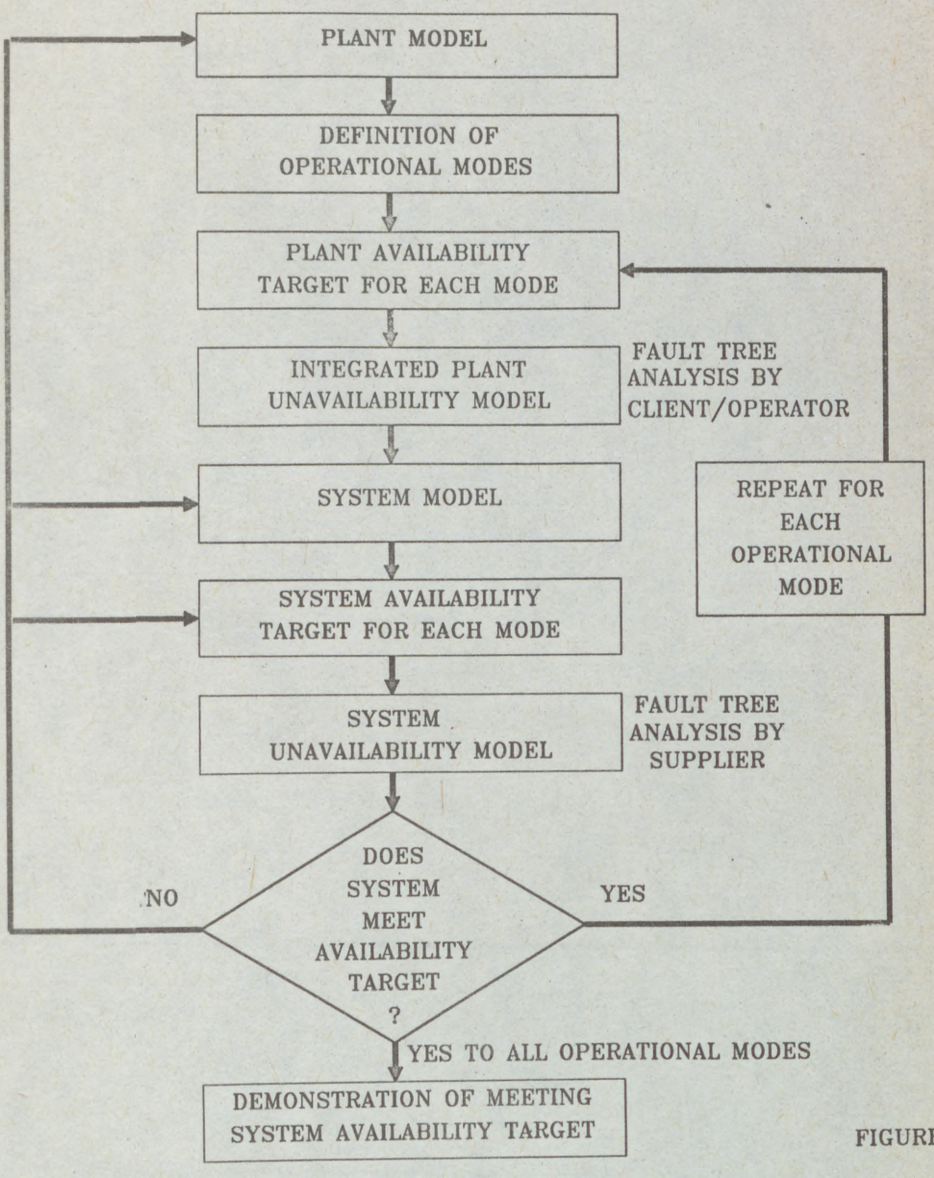


FIGURE 2

LIFE CYCLE COST

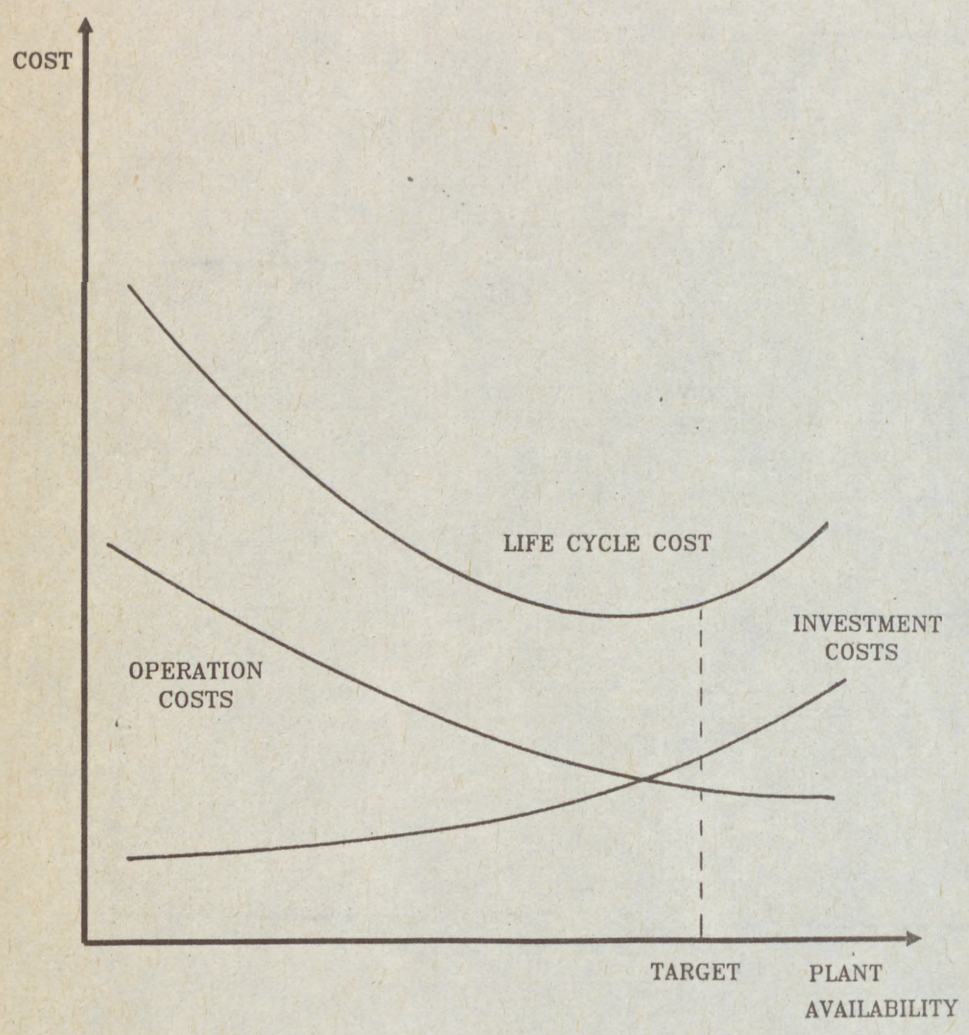
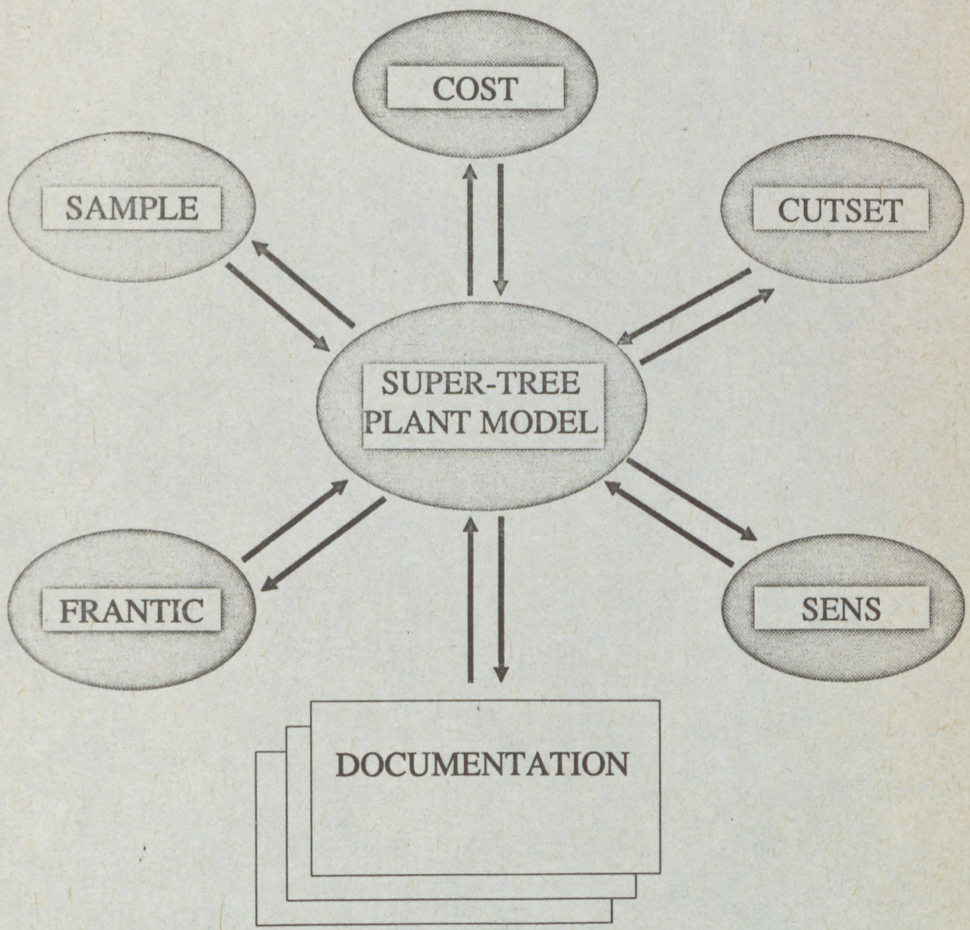
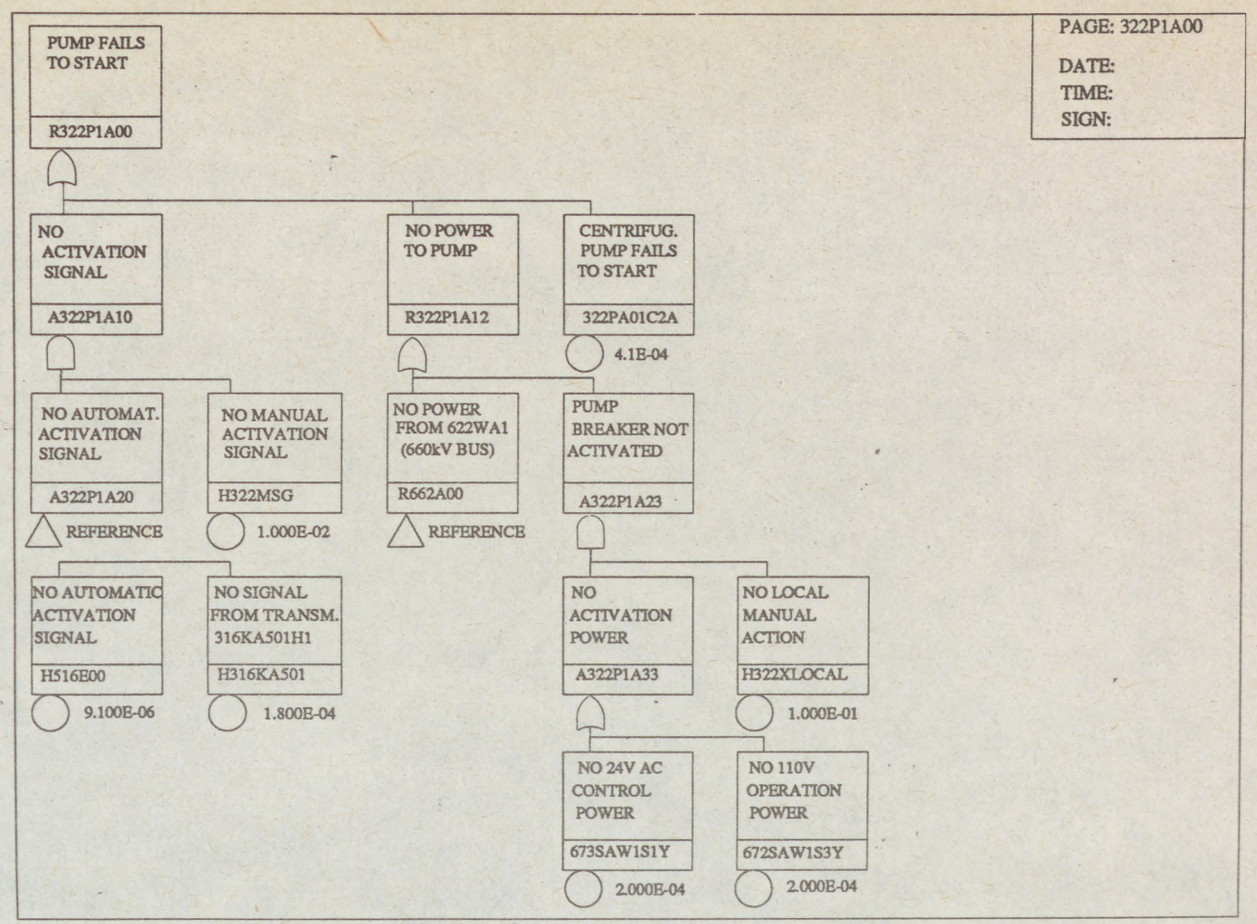


FIGURE 3



The SUPER-NET Structure FIGURE 4

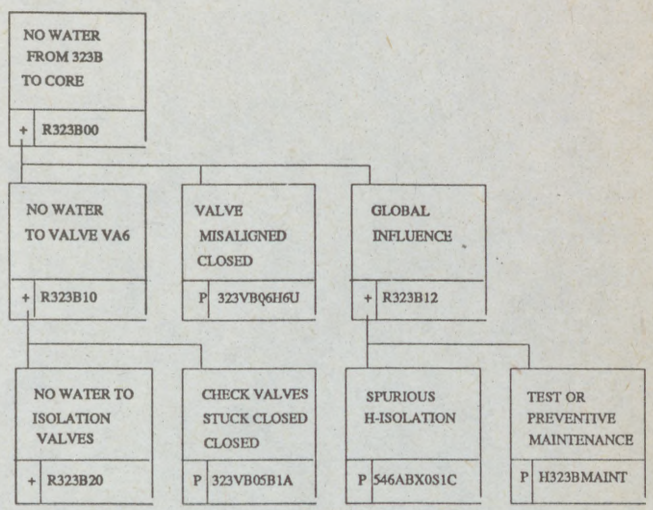


COMPONENT FAULT TREE FOR CENTRIFUGAL PUMP FIGURE 5

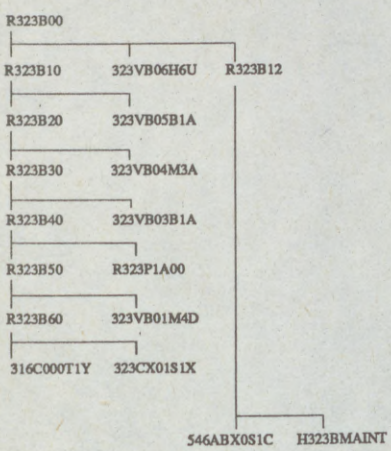
LEVEL 1 / Page Level

TOP EVENT
322A00
322B00
322D00
322CCFA00
322CCFB00
322CCFC00
322CCFD00
322SYSTEM
323A00
323B00
323C00
323D00
323CCFA00
323CCFB00
323CCFC00
323CCFD00
600SYSTEM
:
:
etc

LEVEL 3 / Gate Level



LEVEL 2 / Gate Name Level



LEVEL 4 / Gate Data Menu

Gate name	:323PA01C2A0302
Diamond flag	:N
Diamond mode name	:
Atribute line 1	:B101.3
2	:Inspection, 12,1,2
3	:Overhaul, O.33, 1, 24
Box text	:Pump fails to start
Gate type	:P
Unavailability	:3.0E-03
Time indep unavail	:6.0E-04
Failure rate	6.7E-06
Inspection interval	:30 days
Mean time to repair	:8 hours
Reference	:1.7
Comments	:
Component type	:Centr. pump/stand-by
Failure type	: 3
Data last changed by	:
Data last changed on	:

Handling Levels in SUPER-TREE

FIGURE 6

RELIABILITY ANALYSIS OF STRUCTURAL-COMPLEX SYSTEMS WITH MANY ELEMENTS STATES ON THE BASIS OF MATHEMATICAL AND ELECTRONIC MODELS

Guennady A. VELIGOURSKY, Belorussian Academy of Sciences

USSR

Summary

to G.A.Veligurskiy Report

"Reliability Analysis of Structural-Complex Systems with Many Elements States on the Basis of Mathematical and Electronic Models"

Method of reliability analysis of structural-complex systems with many elements states are considered. The aforementioned methods are based on the use of logical operators mapping onto the logic of the investigated system element state modification. The methods applied are to a sufficient degree formalized and that gives a possibility to perform reliability analysis using computers.

In the report there are considered as well instrumental methods (technical means) of reliability analysis of complex systems, the application of which allows to acquire statistical data on a system state much more quicker and effective than with the usage of computers.

Guennady A. Veligoursky

Reliability Analysis of Structural-Complex Systems with Many Elements States on the Basis of Mathematical and Electronic Models

In the analysis of reliability of complex systems they are usually presented as models which reflect the conditions of a system being in various states depending on the states of its elements. In the majority of such models only two states are taken into account: serviceable and failure states, and the models are illustrated in the form of a diagram or a graph of states. The methods of the reliability factors analysis of complex systems, the models of which are represented in the form of a diagram and the elements of which are capable to assume two states only, is explained in detail in numerous literature. The task of reliability analysis is becoming much more complicated for the so-called structural-complex systems, which are not subjected, in common case, to simplification into successive-parallel combinations (from the point of view of reliability), and the elements of such systems are capable to assume failure states of different types.

For reliability analysis of such systems there exists a proposal to divide the complete set of states into K subsets of states, in which there is present one subset of serviceable states of a system and (K-1)-subsets of failure states. The task consists in the defining of a probability of the system to be found in one of K subsets of states. The evaluation of the system probability to be found in one of K subsets of states implies construction of structural functions (mathematical models) which illustrate the condition of a system being in a certain state depending on its elements states.

Element State Mathematical Model

Let us assume that system S consists of n elements. The elements of the system would mean any device the reliability of which is taken into account irrespective of its structure and the reliability of its components. Let us designate ν-th element of the system with c_ν, $\nu = \overline{1, n}$. Each element is characterized by a finite number of states among which it is possible to reckon different levels of serviceability as well as failures of different types. In the system an element is connected with other elements through its inputs and outputs. That is why the state of an element in the system would be defined by its output state and would depend on its inner and input states. Element input states are characterized by output states of other elements, structurally connected with the element under consideration; element inner states are characterized by various processes of ageing and wear taking place just in the element itself. As a result of such processes the element from the serviceable state transforms into the failure state of different types. Let it be that output, inner and input states of ν-th state at fixed moment of time t are described as variable y_ν, z_ν, x_ν accordingly. The aforementioned variables acquire the values from the set of conditions $\Gamma = \{\gamma_1, \gamma_2, \dots, \gamma_\kappa\}$ Then each element c_ν is capable to acquire κ_ν different conditions from the set Γ, where $\kappa_\nu \leq \kappa$. For simpicity it is implied that $\kappa_\nu = \kappa$.

Let's introduce the notion of the elementary event. For the elementary events for the element c_ν we would understand the events $y_\nu^{(\ell)}$, $x_\nu^{(\ell)}$, $z_\nu^{(\ell)}$, meaning that the corresponding variables y_ν, x_ν, z_ν would acquire the value γ_ℓ, $\ell = \overline{1, \kappa}$. Then

the set of the corresponding elementary events forms a full group of incompatible events. If the probability of each event $Prob\{y_\nu=\Gamma_\ell\}$, $Prob\{x_\nu=\Gamma_\ell\}$, $Prob\{z_\nu=\Gamma_\ell\}$ in denoted by $P_{y_\nu}^{(\ell)}$, $P_{x_\nu}^{(\ell)}$, $P_{z_\nu}^{(\ell)}$ accordingly, than $\sum_{\ell=1}^{\kappa} P_{y_\nu}^{(\ell)}=1$; $\sum_{\ell=1}^{\kappa} P_{x_\nu}^{(\ell)}=1$; $\sum_{\ell=1}^{\kappa} P_{z_\nu}^{(\ell)}=1$.

Let's consider the element model with a set of states. (Fig.I).

Let the element have u outer inputs, ω inner inputs and one output. At a predetermined moment of time t on the outer input j is applied random signal $x_j(t)\in\Gamma$, which corresponds to the output state of the element structurally connected with the element under consideration. These signals determine the input state of the element x_σ. On each inner input every time is applied random signal $x_j(t)$ of strictly defined type, corresponding to one of the Γ set of failure states. These signal determine the inner state of the element z_ν. Than at the predetermined moment of time t on the output of the element there would be signal $y_\nu(t)\in\Gamma$, the value of which is calculated with the help of operator L mapping into various sets of input and inner states from set Γ into set Γ:

$$y_\nu(t)=L(x_1(t), x_2(t), \ldots, x_u(t), z_1(t), z_2(t), z_\omega(t)) \qquad (1)$$

Basis Operators

Each element of the system is charaterized by different output states depending on its inner states and the states of other elements, structurally connected with the element under consideration. Each output state of the element is determined by elementary logical sentence. Logical sentence, which in a unique fashion determines one specific state of the element depending on the inner and input states would be called basis operator. Then, the set of basis operators would be called logical operator of C_ν element. The basis

operators map into all the possible output states of the element under consideration depending on its inner and input states. Output ℓ state in the logical operator is a set of basis operators describing 1-state. By the way, all basis operators forming in the logical operator different output states constitute a full group of incompatible events.

Let us consider basis logical operators in which output state of the element in the fixed moment of time t is determined by the set of input states and one inner state, belonging to the set Γ.

Thus for each basis operator expression (I) is derived as follows

$$y_\nu(t) = L_y(x_1(t), x_2(t), \ldots, x_u(t), z_\nu(t)) \qquad (2)$$

Futher, some basis operators of expression (2) will be derived by the following logical expressions:

$$y_\nu(t) = \gamma_j, \quad \text{if} \quad x_1(t) = \gamma_{\ell_1},$$

$$x_2(t) = \gamma_{\ell_2}, \ldots, x_u(t) = \gamma_{\ell_u} \quad \text{and} \quad z_\nu(t) = \gamma_i \qquad (3)$$

where $j, i, \ell_1, \ell_2, \ldots, \ell_u \in \{1,2,\ldots,\kappa\}$

To simplify this expression parameter t will be omitted.

Apparently, basis operator (3) is complete in the sense that by means of it any expression, characterizing output state of the element depending on various sets of input and inner states, can be derived. Yet, its direct use for plotting the model of the element state is quite a difficult task. Therefore, instead of one universal basis operator it is expediend to plot a limited number of special basis operators, each of which corresponds to definite logic expressions, characterizing the principle of the real element functioning.

Let us consider the main basis operators, by means of which the models of various states of complex systems elements are plotted.

1. Operator of inner state. For this operator the output state of the element is determined only by its inner state, i.e. it depends on a casual value z_ν.

Thus

$$y_\nu = \gamma_j, \quad \text{if} \quad z_\nu = \gamma_i \tag{4}$$

2. Operator of input and inner states. For this operator output state of the element is determined by the inner state $z_\nu = \gamma_i$ and by the availability of at least one state $x_{\nu_t} = \gamma_\ell$ at any of its inputs, i.e.

$$y_\nu = \gamma_j, \text{ if } \bigvee_{t=1}^{u} (x_{\nu_t} = \gamma_\ell) \text{ and } z_\nu = \gamma_i \tag{5}$$

3. Operator $\bigsqcup_{max}$ (maximum)

$$y_\nu = \gamma_j, \text{ if } \max(x_1, x_2, \ldots, x_u) = \gamma_\ell \text{ and } z_\nu = \gamma_i \tag{6}$$

4. Operator $\bigsqcup_{min}$ (minimum)

$$y_\nu = \gamma_j, \text{ if } \min(x_1, x_2, \ldots, x_u) = \gamma_\ell \text{ and } z_\nu = \gamma_i \tag{7}$$

5. Operator of exclusion due to its input states is characterized by the equaility, but one, of the input states of the element, i.e. $y_\nu = \gamma_j$, if $x_1 = x_2 = \ldots = x_{t-1} = x_{t+1} = \ldots = x_u = \gamma_\ell$ and there is

$$x_t = \gamma_s \quad \text{and} \quad z_\nu = \gamma_i, \quad i, j, s, \ell \in \{1, 2, \ldots, \kappa\}, \quad t = \overline{1, u} \tag{8}$$

6. Operator of coincidence

$$y_\nu = \gamma_j, \quad \text{if} \quad x_1 = x_2 = \gamma_\ell \quad \text{and} \quad z_\nu = \gamma_i \tag{9}$$

This operator is applied to the elements of the comparison type. In the particular case the serviceable state is always observed at the operator output if the two signals, corresponding to the same state at the outputs of the other units, are transferred to its in-

puts, i.e.

$$y_\nu = \gamma_\kappa \qquad \text{if} \quad x_1 = x_2 = \gamma_\ell \quad \text{and} \quad z_\nu = \gamma_\kappa$$

7. Operator of different inputs

$$y_\nu = \gamma_j \quad , \text{ if all input signals are pairwise different and if } \quad x_\nu = \gamma_i \tag{10}$$

Apparently, this operator is important if the number of inputs is less than κ or is equal to it. In the particular case, if $u = 2$ an operator inverse to the operator of coincidence will be obtained.

Plotting of Structural Functions of Complex Systems States

In order to plot structural functions, it is necessary to regard the present system as a diagram, where the points of physical joints or the elements (units) of the real system correspond to the multiplicity of peaks V, while the communication lines between them correspond to the set of edges E. Thus, each peak of the net posesses the logical operator, characterizing the logics of transfer of the present element from one state into another. If logical operator is applied to the point of physical joint of the system, it inner state is considered to be absolutely safe, i.e. $z_\nu = \gamma_\kappa$. Therefore, output state of the present operator will be determined by the output states of the elements, structurally connected with the present elements.

For the system as a whole let us introduce a random variable Y_s, which may transfer to any condition within the set Γ.

Then the state of the system Y_s in the moment of time t is synonymously determined by the variables

and can be derived as follows

$$Y_s(t) = Y_s(y_1(t), y_2(t), \ldots, y_n(t)) = Y_s(\vec{y}(t))$$

Let us assume that $Y_s(t)$ is the structural function of the system states. Usually $Y_s(t)$ is the functional link between the output state of the system and the output states of the elements. Structural function of states determines the split of sets of all n-parameter vectors with κ-state of the elements $E = \{\vec{y}(t)\}$ into κ sub-sets $E_1 = \{\vec{y}(t)\}: Y_s(t) = r_1\}, \ldots, E_{\kappa-1} = \{\vec{y}(t): Y_s(t) = r_{\kappa-1}\}$, $E_\kappa = \{\vec{y}(t): Y_s(t) = r_\kappa\}$.

Moreover,

$$Y_s(t) = \begin{cases} r_1, & \text{if the system is characterized by the state of failure of the first type} \\ \ldots\ldots\ldots & \\ r_{\kappa-1}, & \text{if the system is characterized by the state of failure of the } (\kappa-1)\text{-type} \\ r_\kappa, & \text{if the system is characterized by the serviceable state} \end{cases}$$

where $Y_s(t) \in \Gamma = \{r_1, r_2, \ldots, r_\kappa\}$

In future in order to make the expressions shorter the variables will be omitted.

The aforeintroduced notion of logical operator provides formalizing of the plotting process of the structural functions of the system states. Let us consider structural functions of states via an event.

This event, characterized by $Y_s = r_\ell$, will be expressed in the form of $Y_s^{(\ell)}$, $\ell = \overline{1,\kappa}$, while the probability of the event

$\mathrm{Prob}\{Y_s = \ell\}$ in the form of $P_s^{(\ell)}$, where $\sum_{\ell=1}^{K} P_s^{(\ell)} = 1$. This sub-set of states of the system $E_\ell \subset E$ is unification. $Y_s^{(\ell)} = \bigcup_{j=1}^{m} Y_j^{(\ell)}$, where $Y_s^{(\ell)}$ is determined by means of recurrent relations, which in accordance with the model of the system, provided in the form of the net and the corresponding logical operators consequently determine output states of the elements from the output of the net towards its input, i.e.

$$Y_s^{(\ell)} = Y_1^{(\ell)} \cup Y_2^{(\ell)} \cup \ldots \cup Y_m^{(\ell)}, \qquad \ell = \overline{1,K} \tag{11}$$

where $Y_j^{(\ell)} = y_{j_1}^{(\ell_1)} \cdot y_{j_2}^{(\ell_2)} \cdot \ldots \cdot y_{j_s}^{(\ell_s)} = \bigwedge_{t=1}^{s} y_{j_t}^{(\ell_s)}$, $\ell_t = \{1,2,\ldots,K\}$

is the minimum number of elementary events, the availability of which provides ℓ-state of the system.

$Y_s^{(\ell)}$ plotting is carried out by means of logical operators which can be expressed in this case in the form of the finite number of elementary events. Since the logical operators are plotted on the basis of the basis operators, the latter can also be expressed via elementary events of the corresponding structural functions.

For the basis operator I, for example

$y_\nu^{(j)} = z_\nu^{(i)}$, $j, i \in \{1,2,3,4\}$

In turn the basis operator 3 can be expressed:

$$y_\nu^{(j)} = z_\nu^{(i)}(x_1^{(\ell)} + x_2^{(\ell)} + \ldots + x_u^{(\ell)})(x_1^{(1)} + \ldots + x_1^{(\ell)})(x_2^{(1)} + \ldots + x_2^{(\ell)}) \ldots (x_u^{(1)} + \ldots + x_u^{(\ell)}), \tag{11'}$$

where $\ell = \max\{\ell_1, \ell_2, \ldots, \ell_u\}$

When $\ell = 4$

$$y_\nu^{(j)} = (x_1^{(4)} + x_2^{(4)} + \ldots + x_u^{(4)})\, z_\nu^{(i)} \tag{11''}$$

When $\ell = 3$

$$y_\nu^{(j)} = (x_1^{(3)} + \ldots + x_u^{(3)})(x_1^{(1)} + x_1^{(2)} + x_1^{(3)})(x_2^{(1)} + x_2^{(2)} + x_2^{(3)}) \cdot \ldots \cdot (x_u^{(1)} + x_u^{(2)} + x_u^{(3)}) z_u^{(i)},$$

when $\ell = 2$

$$y_\nu^{(j)} = (x_1^{(2)} + x_2^{(2)} + \ldots + x_u^{(2)})(x_1^{(1)} + x_1^{(2)})(x_2^{(1)} + x_2^{(2)}) \ldots \ldots (x_u^{(1)} + x_u^{(2)}) z_\nu^{(i)},$$

when $\ell = 1$

$$y_\nu^{(j)} = x_1^{(1)} \cdot x_2^{(1)} \cdot \ldots \cdot x_u^{(1)} \cdot z_\nu^{(i)}$$

Other operators are represented in the same way.

Representation of basis operators with the help of elementary events has given a possibility of plotting the structural functions of the entire system.

The probability of the system being in ℓ state $Y_s^{(\ell)}$ is determined from the formula

$$P_s^{(\ell)} = Prob\{Y_s = Y_\ell\} = Prob\{Y_1^{(\ell)} \cup Y_2^{(\ell)} \cup \ldots \cup Y_m^{(\ell)}\} \qquad (12)$$

Probability calculations of the ℓ state of the system from the formula of the sum of events probabilities is a difficult task, especially if the number of members is great enough. (see Fig 2)

The main task of the probabilities $P_s^{(\ell)}$ calculations consists in reduction of the set of compatible events to the equivalent set of incompatible events $F_j^{(\ell)}$. Then, probability $P_s^{(\ell)}$ would be found from the formula

$$P_s^{(\ell)} = Prob\{\bigcup_{j=1}^{m} Y_j^{(\ell)}\} = \sum_{j=1}^{n} Prob\{F_j^{(\ell)}\} \qquad (13)$$

where $Prob\{F_j^{(\ell)}\}$ - is the product of probabilities of

elementary events of $F_j^{(\ell)}$. For the transformation of compatible events into incompatible events we would use the method of orthogonalization, which consists of the following.

Let the structural function of the type (11) represent disjunctive normal form of elementary conjunctions Y_j , that is

$$Y_s = Y_1 + Y_2 + \dots + Y_m \qquad (14)$$

where: Y_j , $j = \overline{1,m}$ have numerals according to increase of their power.

Let's determine relative complement D_{ij} between members Y_i and Y_j of the sentence (14):

$$D_{ij} = Y_i \setminus Y_j = y_1 y_2 \dots y_q \; ; \; i = \overline{1, j-1} \; , \; j = \overline{2,m}$$

Here, D_{ij} is the conjunction of those variables which are not present in Y_j . The acquired conjunctions for fixed j we would arrange in an ordered fashion in accordance with the increase of their powers. After that we would eliminate those D_{ij} , for which there is present such D_{ℓ_j} , that $D_{\ell_j} \subset D_{ij}$. For each Y_j from the left $i_j \leq j-1$ conjunctions $D_{i_1 j}, D_{i_2 j}, \dots, D_{i_j j}$ their negation is constructed in accordance with the following rule:

$$\overline{D}_{ij} = \overline{y}_1 + y_1 \overline{y}_2 + \dots + y_1 y_2 \dots \overline{y}_q \qquad (15)$$

Let's write down recurrent expression for Φ_j, $j = \overline{2,m}$, where Φ_m is a logical function, orthogonality Y_s of the expression (14):

$$\Phi_1 = Y_1 \; , \quad \Phi_j = \Phi_{j-1} + \overline{D}_{i_1 j} D_{i_2 j} \dots \overline{D}_{i_j j} Y_j = \Phi_{j-1} + R_j Y_j$$

where $R_j = \overline{D}_{i_1 j} \dots \overline{D}_{i_j j}$ $(j = \overline{1,m})$, $R_1 \equiv 1$.

And for Φ_m we would come up to:

$$\Phi_m = Y_1 + R_2 Y_2 + \ldots + R_m Y_m$$

or $\Phi_m = F_1 + F_2 + \ldots + F_n$

where $F_1 = Y_1$, $F_i = R_i Y_i$, $i \neq 1$.

It's not difficult to show that $\Phi_m = Y_s$ and all the members in Φ are pairwise orthogonal. Then each variable of the elementary conjunction F_j is changed for corresponding probability and the calculation of $P_s^{(\ell)}$ is performed from the formula (13), as an ordinary sum of events probabilities.

Thus, we have considered the method of calculation of the probability of a system being in ℓ state with elements in many states, when logical operators were used as mathematical models of elements states.

Reliability analysis of structural-complex systems may be more effective (from the point of view of time expenditure) if we use special purpose stochastic units, in which as elements models electron models are used, mapping mathematical models into logical operators. Reliability analysis of complex systems using special purpose stochastic units has got the name - instrumental methods. During reliability analysis using stochastic installation there is performed physical model simulation of the system under consideration. Physical model of a complex system is a model in which the points of physical jointing (the points of crossing) of the real system and the points of physical jointing of the model correspond; in which the blocks of the real system correspond to the blocks of the physical model states; communication lines between the blocks of the real system correspond to the communication lines between the physical model state blocks. State blocks of the physical model are specially designed electronic circuits on the digital equipment ele-

ments, output states of which are determined in accordance with the logical operator mapping into the logics of the system real block state change.

So as the physical model of the system under consideration is an electronic model, then any changes in its structure immidiately find its representation in the model's output.

In the common case the method of physical modelling of the complex systems states consists in the fact that in the physical model of a system, which is continously in compressed time scale, there are simultaneously being generated random values of time moments for appearance of various failure states of all the elements of the system with the predetermined distribution functions and random processes of parameter changes of these elements on time basis. The process of random values and random process simulation is performed in uniquie time scale.

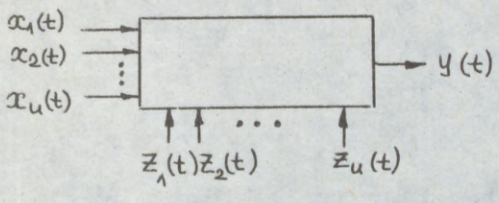
The basis of instrumental methods of complex systems reliability analysis, as well as when using analytical methods and methods of computer statistical modelling, is made of logical operators, determining each element state in accordance with its inner state, other elements states and the time of. (see Fig 3)

Fig. 4 is an electrical circuit diagram of the stochastic installation. On the jack field of the stochastic installation there are placed input, output and internal terminals of each states block. In accordance with the structure of the system under consideration, interconnecting inputs and outputs of states block, its model is being composed on the jack field. To the internal inputs of states block are connected the outputs of controlled probabilities converter, on which the necessary functions of random values distribution

are composed. In the analysis of complex systems, taking into account gradual failures, to the internal inputs are connected the corresponding random processes generator. The installation is capable to function in the mode of reliability analysis of restoring and non-restoring systems.

Table I presents the results of complex system failures statistical modelling (see Fig. 5) using up-to-date computer and stochastic installation. As you can see from the Table, the results of the modelling are commensurable. Nevertheless, operative effect on the system physical model in the process of the analysis of its reliability (changes in the structure, input of new initial data), as well as the rate of acquiring the results of modelling demonstrate the advantage of stochastic installation (in reliability analysis of structural-complex systems) in comparison with modern computers.

Fig. 1. Plotting of the System Element Logical Operator



Element model

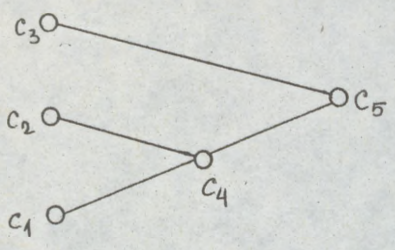
Let's consider plotting of the logical operator, using an example of a specific element. Failure detector would be chosen as an example of the specific element and the states of which are defined by the following logical sentence. Failure detector is in serviceable state $y_v = \gamma_4$, if on its two inputs are applied two equal signals, conformable either to serviceable states of other elements output signals, structurally connected with the element under consideration, or to failure states of these elements, and failure detector according to its inner state is in serviceable state $z_v = \gamma_4$. Failurd detector is in a failure state $y_v = \gamma_1$, if on its inputs are applied signals, conformable to different states and the failure detector itself is in serviceable state. Futhermore, failure detector may be in the state of false switching-off γ_2 and state of failure to operate γ_3. In such a case, do not depending on input signals, on the output of failure detector there would always be state γ_1 and γ_4. Thus, logical operator for failure detector would have the following form:

$$y_{\Delta o} = \begin{cases} r_4, \text{ if } & x_1 = x_2 = r_4 \text{ and } z_{\Delta o} = r_4 \\ r_4, \text{ if } & x_1 = x_2 = r_1 \text{ and } z_{\Delta o} = r_4 \\ r_1, \text{ if } & x_1 \neq x_2 \text{ and } z_{\Delta o} = r_4 \\ r_4, \text{ if } & z_{\Delta o} = r_3 \\ r_1, \text{ if } & z_{\Delta o} = r_2 \end{cases}$$

To each of the presented logical sentences correspond (in sequence) the following basis operators: 6, 6, 7, 1, 1. Then, the logical operator may be expressed in the following form:

$$y_{\Delta o} = \begin{cases} 6 & j=4 & i=4 & \ell=4 \\ 6 & j=4 & i=4 & \ell=1 \\ 7 & j=1 & i=4 & \\ 1 & j=4 & i=3 & \\ 1 & j=1 & i=2 & \end{cases}$$

Fig. 2. Plotting the System Structural Functions



System model

Let we have system model shown as a net on the picture. Let the inner states of the elements c_1 , c_3 , c_5 are defined by two states, i.e. z_1 , z_3 , $z_5 \in \{r_1, r_4\}$ and the elements c_2, c_4 - by four states, i.e. $z_2, z_4 \in \{r_1, r_2, r_3, r_4\}$ Let's write for each element their logical operators. So as elements c_1 , c_2 , c_3 are input elements, their output states are determined only by their inner states. Thus, for c_1, c_3 we have:

$$y_{1(3)} = \begin{cases} r_1, & \text{if } z_{1(3)} = r_1 \quad 1 \quad j=1, \; i=1 \\ r_4, & \text{if } z_{1(3)} = r_4 \quad 1 \quad j=4, \; i=4 \end{cases} \qquad (1)$$

for c_2

$$y_2 = \begin{cases} r_1, \text{if } z_2 = r_1 \quad 1 \quad j=1 \quad i=i \\ r_2, \text{if } z_2 = r_2 \quad 1 \quad j=2 \quad i=2 \\ r_3, \text{if } z_2 = r_3 \quad 1 \quad j=3 \quad i=3 \\ r_4, \text{if } z_2 = r_4 \quad 1 \quad j=4 \quad i=4 \end{cases} \qquad (2)$$

for element c_4 the logical operator will be written in the form:

$$y_4 = \begin{cases} f_1, \text{ if } \max(x_1, x_2) = f_1 & \text{and } z_4 = f_4 \;\; 3 \;\; j=1, \ell=1, i=4 \\ f_2, \text{ if } \max(x_1, x_2) = f_2 & \text{and } z_4 = f_4 \;\; 3 \;\; j=2, \ell=2, i=4 \\ f_3, \text{ if } \max(x_1, x_2) = f_3 & \text{and } z_4 = f_4 \;\; 3 \;\; j=3, \ell=3, i=4 \\ f_4, \text{ if } \max(x_1, x_2) = f_4 & \text{and } z_4 = f_4 \;\; 3 \;\; j=4, \ell=4, i=4 \\ f_1, \text{ if } z_4 = f_1 & 1 \;\; j=1, i=1 \\ f_2, \text{ if } z_4 = f_2 & 1 \;\; j=2, i=2 \\ f_3 \text{ if } z_4 = f_3 & 1 \;\; j=3, i=3 \end{cases}$$

For element c_5 the logical operator would acquire the form:

$$y_5 = \begin{cases} f_1, \text{ if } \min(x_1, x_2) = f_1 & \text{and } z_5 = f_4 \;\; 4 \;\; j=1, \ell=1, i=4 \\ f_2, \text{ if } \min(x_1, x_2) = f_2 & \text{and } z_5 = f_4 \;\; 4 \;\; j=2, \ell=2, i=2 \\ f_3, \text{ if } \min(x_1, x_2) = f_3 & \text{and } z_5 = f_4 \;\; 4 \;\; j=3, \ell=3, i=4 \\ f_4, \text{ if } \min(x_1, x_2) = f_4 & \text{and } z_5 = f_4 \;\; 4 \;\; j=4, \ell=4, i=4 \\ f_1, \text{ if } z_5 = f_1 & 1 \;\; j=1, i=1 \end{cases}$$

To the right of the logical operators is given their formalized form. Numerals point the number of a logical operator, indices ℓ, i, j - input, inner and output values of element states conformable to the given basis operator. Thus, the logical operator of each element may be expressed with the help of the according elementary event using the sentences (4) - (10-. Let's consider the process of the system structural function plotting, accuring in its serviceable state. To the output state of the system corresponds logical operator (4), for which due to basis operators we have

$$Y_5^{(4)} = y_5^{(4)} = x_1^{(4)} x_2^{(4)} z_5^{(4)}$$

In the given sentence input variables x_1, x_2 of element c_5 correspond to output variables y_3, y_4 of elements c_3, c_4 ,

structurally connected with c_5 . That's why $x_1^{(4)} = y_3^{(4)}$ and $x_2^{(4)} = y_4^{(4)}$. Futher, instead of input variables of the element substitute corresponding to them output variables of the elements which are structurally connected with the element under consideration. Then, $Y_5^{(4)}$ will become $Y_5^{(4)} = y_3^{(4)} y_4^{(4)} z_5^{(4)}$. As the output state of element c_3 depends only upon its inner state, then $y_3^{(4)} = z_3^{(4)}$, and in this case

$$Y_5^{(4)} = y_4^{(4)} z_3^{(4)} z_5^{(4)} \qquad (5)$$

Output state of element c_4 corresponding to serviceability state $y_4^{(4)}$ is describex by basis operator 3 and in accordance with (11') and (11"), it will be wtitten down in the following form:

$$y_4^{(4)} = (y_1^{(4)} + y_2^{(4)}) z_4^{(4)}$$

As the output state of elements c_1 and c_2 depends only upon inner states, then $y_1^{(4)} = z_1^{(4)}$ and $y_2^{(4)} = z_2^{(4)}$. That's why

$$y_4^{(4)} = (z_1^{(4)} + y_2^{(4)}) z_4^{(4)}$$

Introducing the given expression in (5), we shall get the structural function of the occurring of the system, illustrated in the picture in the form of a net, in its serviceable state

$$Y_5^{(4)} = (z_1^{(4)} + z_2^{(4)}) z_4^{(4)} z_3^{(4)} z_5^{(4)} = z_1^{(4)} z_4^{(4)} z_3^{(4)} z_5^{(4)} + z_2^{(4)} z_4^{(4)} z_3^{(4)} z_5^{(4)} .$$

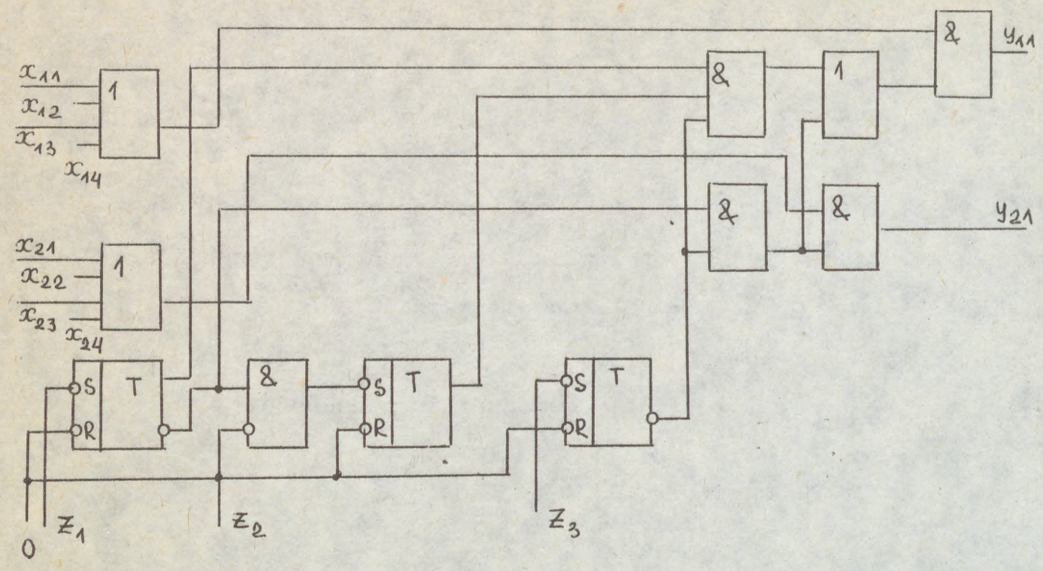


Fig. 3 States Block

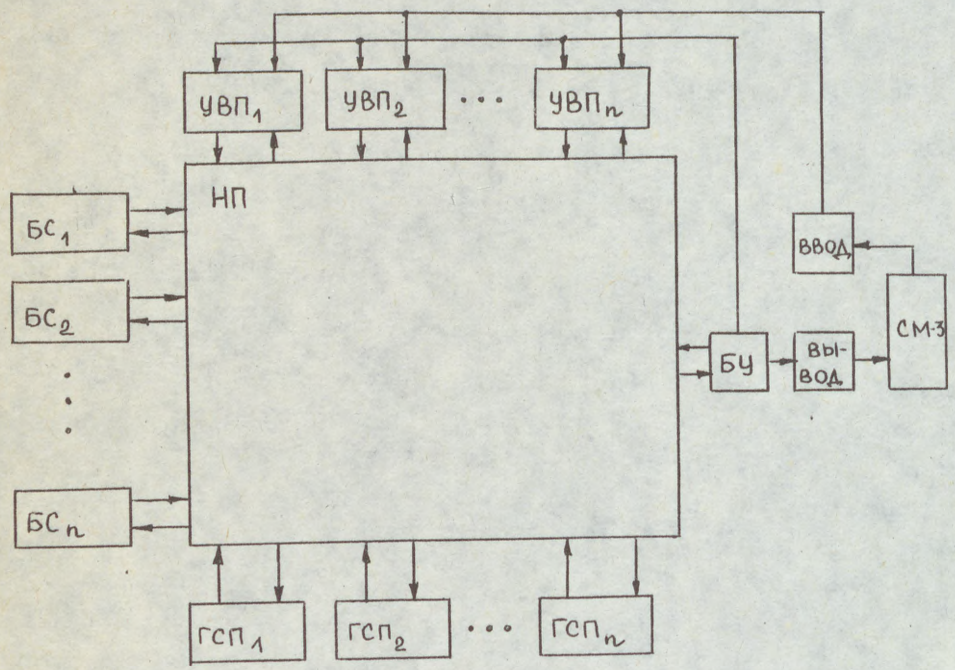


Fig. 4 Stochastic installation

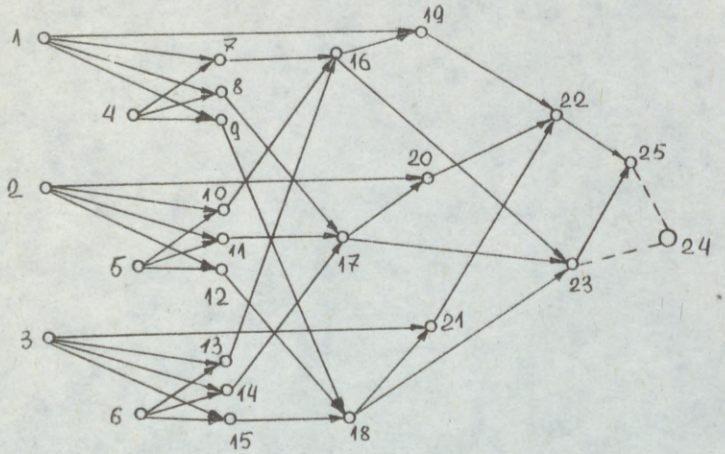


Fig. 5 System model in the form of a network

Table 5

System State Probability	: Modelling, Using Computer	: Modelling, Using Stochastic Installation
Controlled Failure	0,0014417	0,0015135
Detected Failure	0,0111083	0,0111698
False Failure	0,0050467	0,0050596
Serviceable State	0,9824033	0,9822571

ESTIMATION OF PRODUCT RELIABILITY USING FUZZY LIFE TIME DATA

Reinhard VIERTL, Technische Universität Wien

Austria

Estimation of Product Reliability using Fuzzy Life Time Data

by

Reinhard VIERTL

Technische Universität Wien

Abstract: In conventional reliability analysis observed life time data are considered as real numbers. These exact data are used for statistical inference in classical or Bayesian way. Real data are usually not exact numbers but fuzzy by the measurement process. This kind of uncertainty is different from statistical variation. Therefore life time data need to be described in a suitable model. One possibility is the description of real life times by so called fuzzy numbers which are generalizations of indicator functions. Statistical inference methods can be adapted to develop an inference method for fuzzy data in reliability estimation. Classical nonparametric and parametric estimation as well as Bayesian estimation procedures using the information from imprecise life time data are explained.

1. Imprecise Life Time Data

In usual statistical reliability estimation observed life times are considered as nonnegative numbers. But real life time data are not exact real numbers $x \geq 0$ but more or less imprecise, also called fuzzy.

This uncertainty is different from random variation of life times described by a stochastic quantity T with corresponding probability model described by the reliability function R(.) defined by

$$R(t) = Pr\{T > t\} \quad \text{for all} \quad t \geq 0 \ .$$

Imprecise life time data can be described suitably by so called fuzzy numbers, i.e. generalizations of real numbers and indicator functions $I_B(.)$ with $B \subseteq \mathbb{R}$. A fuzzy number $\varphi(.)$ is a measurable real function obeying

$$0 \leq \varphi(x) \leq 1 \quad \text{for all} \quad x \in \mathbb{R}$$

and

$$\{x \in \mathbb{R} : \varphi(x) = 1\} \neq \emptyset$$

such that $\{x \in \mathbb{R} : \varphi(x) > 0\}$ is an interval. The function $\varphi(.)$ is also called characterizing function of the fuzzy number. For life times nonnegative fuzzy numbers are of interest, i.e. fuzzy numbers $\varphi(.)$ with

$$\varphi(x) = 0 \quad \text{for all} \quad x < 0 \ .$$

An imprecise observation x^* of the stochastic quantity T, describing the life time of a product, is modelled by a nonnegative fuzzy number $\varphi(.)$. The idealization of an exact life time observation $t \geq 0$ is the special case

$$\varphi(.) = I_{\{t\}}(.)$$

where $I_{\{t\}}(.)$ denotes the one-point indicator function defined by

$$I_{\{t\}}(x) = \begin{cases} 1 & \text{for } x = t \\ 0 & \text{for } x \neq t \,. \end{cases}$$

A general fuzzy number is depicted in figure 1.

Figure 1: Fuzzy Number

2. Fuzzy Data and Sample Information

Let n independent fuzzy observations $x_1^*, \ldots, x_n^*$ of the life time T of a product be available. In abbreviation we write D^* for this sample of n observations, called fuzzy data $D^* = (x_1^*, \ldots, x_n^*)$. The fuzziness in the

i-th observation x_i^* is described by a corresponding characterizing function $\varphi_i(.)$ defining a fuzzy number.

The information in the fuzzy sample $D^* = (x_1^*, \ldots, x_n^*)$ can be used for statistical inference on the life time distribution of the product in different ways.

3. Nonparametric Estimation of the Reliability Function

For exact life time data $D = (x_1, \ldots, x_n)$ the reliability function $R(.)$ is estimated by the empirical reliability function $\hat{R}_n(.)$ defined by

$$\hat{R}_n(x) = \frac{1}{n} \sum_{i=1}^{n} I_{(x,\infty)}(x_i) \quad \text{for all} \quad x \geq 0 \ .$$

For fuzzy data $D^* = (x_1^*, \ldots, x_n^*)$ a generalization of the empirical reliability function can be constructed which takes care of the fuzziness in the data. This generalized empirical reliability function $R_n^*(x | x_1^*, \ldots, x_n^*)$ is a multi-valued real function limited by two functions $R_l(.)$ and $R_u(.)$ defined in the following way. First we assume that the characterizing functions $\varphi_i(.)$ of the fuzzy life times x_i^* have pairwise disjoint supports

$$\text{supp } \varphi_i(.) = \{x \in \mathbb{R} : \varphi_i(x) > 0\} \ .$$

Then the observations can be ordered and we obtain the ordered fuzzy sample $x_{(1)}^*, \ldots, x_{(n)}^*$ with corresponding

characterizing functions $\varphi_{(1)}(.),\ldots,\varphi_{(n)}(.)$. In the interval supp $\varphi_{(i)}(.)$ the functions $R_l(.)$ and $R_u(.)$ are defined in the following way.

$$R_u(x) = \begin{cases} \frac{n-i+1}{n} & \text{for } x \in \{x : \varphi_{(i)}(x)\uparrow \vee \varphi_{(i)}(x) = 1\} \\ \frac{n-i+\varphi_{(i)}(x)}{n} & \text{for } x \in \{x : \varphi_{(i)}(x)\downarrow\} \end{cases}$$

$$R_l(x) = \begin{cases} \frac{n-i+1-\varphi_{(i)}(x)}{n} & \text{for } x \in \{x : \varphi_{(i)}(x)\uparrow\} \\ \frac{n-i}{n} & \text{for } x \in \{x : \varphi_{(i)}(x) = 1 \vee \varphi_{(i)}(x)\downarrow\} \end{cases}$$

and the fuzzy empirical reliability function (FERF) covers the area between $R_l(.)$ and $R_u(.)$ in the interval supp $\varphi_{(i)}(.)$. This is depicted in Figure 2.

Figure 2: Fuzzy Empirical Reliability Function

In the interval supp $\varphi_{(i)}(.)$ the FERF covers the area between $R_l(.)$ and $R_u(.)$

For interval data, i.e. $x^*_{(i)}$ has characterizing function $\varphi_{(i)}(.) = I_{(a,b)}(.)$ this FERF has an immediate practical interpretation. The area between the functions $R_l(.)$ and $R_u(.)$ covers the possible empirical reliability functions between the most pessimistic and most optimistic precise interpretations of the observed life times. This is explained in Figure 3.

Figure 3: Fuzzy Empirical Reliability Function
for Interval Life Time Data

For fuzzy life time data with intersecting support of the corresponding characterizing functions a superposition of the above construction is possible.

4. Parametric Life Time Distributions

For parametric life time distribution model $T \sim f(.|\theta)$ with density $f(.|\theta)$ and parameter space Θ the information

in a fuzzy sample $x_1^*,\ldots,x_n^*$ with corresponding characterizing functions $\varphi_i(.)$ for $i=1,\ldots,n$ is used for statistical inference in the following way. The combined fuzzy sample is a fuzzy vector in $[0,\infty)^n$ defined by

$$\varphi(x_1,\ldots,x_n) = \prod_{i=1}^{n} \varphi_i(x_i) \quad \text{for} \quad x_i \geq 0$$

in case of independent observations. In case of dependent observations also other combinations to obtain a combined fuzzy sample are possible.

For a vector $(x_1,\ldots,x_n) \in \mathbb{R}^n$ we use the short notation $\underline{x}$, i.e. $\underline{x} = (x_1,\ldots,x_n)$ and for the combined fuzzy sample $\varphi(\underline{x})$, i.e.

$$\varphi(\underline{x}) = \varphi(x_1,\ldots,x_n) .$$

5. Classical Parametric Estimation for Fuzzy Life Time Data

Assuming a parametric family of life time distributions with parameter θ classical estimators $\hat{\vartheta}(T_1,\ldots,T_n)$ for sample $T_1,\ldots,T_n$ of the life time T can be adapted for fuzzy samples $x_1^*,\ldots,x_n^*$ in the following way. If the corresponding characterizing functions are $\varphi_1(.),\ldots,\varphi_n(.)$, a fuzzy estimator θ^* for the parameter θ is obtained using the combined fuzzy sample $\varphi(\underline{x})$ from section 4. The fuzzy estimator θ^* is the fuzzy subset $\xi(.)$ of the parameter space Θ defined by

$$\xi(\theta) = \sup \{ \varphi(\underline{x}) : \mathfrak{J}(\underline{x}) = \theta \} .$$

A fuzzy subset of Θ is characterized by a real valued and normalized measurable function on Θ which means a generalization of indicator functions. For exact life time data $t_i \hat{=} I_{\{t_i\}}(.)$ this definition specializes to the traditional point estimator $\mathfrak{J}(t_1,\ldots,t_n)$ for θ. The sup - operation guarantees a conservative estimation result.

This concept allows also a generalization of the construction of confidence regions for the parameter θ. Let $\kappa(T_1,\ldots,T_n)$ be a classical confidence function for θ with confidence level $1-\alpha$. Then the concrete confidence region for exact life time data $x_1,\ldots,x_n$ is denoted by $\kappa(\underline{x})$. This confidence set can be represented by the indicator function $I_{\kappa(\underline{x})}(.)$ defined on the parameter space Θ by

$$I_{\kappa(\underline{x})}(\theta) = \begin{cases} 1 & \text{for } \theta \in \kappa(\underline{x}) \\ 0 & \text{for } \theta \notin \kappa(\underline{x}) . \end{cases}$$

For fuzzy data with corresponding characterizing functions $\varphi_1(.),\ldots,\varphi_n(.)$ a fuzzy confidence region Θ^* for θ is obtained as a fuzzy subset of Θ using the combined fuzzy sample $\varphi(\underline{x})$. For confidence level $1-\alpha$ using the confidence function $\kappa(\ldots)$ for exact samples the fuzzy subset Θ^* of Θ is determined by the characterizing function $\xi(.)$ defined

by $\xi(\theta) = \sup\{\varphi(\underline{x}) : \theta \in \kappa(\underline{x})\}$.

This is a conservative generalization of exact confidence regions by

$$\xi(\theta) = 1 \quad \text{for} \quad \theta \in \cup\{\kappa(\underline{x}) : \varphi(\underline{x}) = 1\} .$$

For exact data this definition reduces to the standard confidence region.

6. Bayes' Theorem for Fuzzy Life Times

In Bayesian inference for parametric life time model $T \sim f(.|\theta)$, exact life time data $D = (t_1,\ldots,t_n)$, and a-priori distribution $\pi(\theta)$ of the stochastic quantity $\tilde{\theta}$ describing the uncertainty concerning θ the a-posteriori distribution (density) $\pi(\theta|D)$ of $\tilde{\theta}|D$ is calculated via Bayes' theorem

$$\pi(\theta|D) \propto \pi(\theta).l(\theta;D) .$$

For fuzzy data $D^* = (x_1^*,\ldots, x_n^*)$ with corresponding characterizing functions $\varphi_1(.),\ldots,\varphi_n(.)$ Bayes' theorem can be generalized in order to obtain a fuzzy analogue $\pi^*(\theta|D^*)$ of the a-posteriori distribution using the combined fuzzy sample $\varphi(\underline{x})$ from section 4. For every $\underline{x} \in [0,\infty)^n$ the a-posteriori density $\pi(\theta|\underline{x})$ is calculated via Bayes' theorem. By variation of $\underline{x}$ according to $\varphi(\underline{x})$

for fixed θ a fuzzy number $\psi_\theta(.)$ is obtained by

$$\psi_\theta(y) = \sup\{\varphi(\underline{x}) : \pi(\theta|\underline{x}) = y\},$$

representing the fuzzy value of the a-posteriori density for argument θ. The whole fuzzy a-posteriori distribution for fuzzy data is the family $(\psi_\theta(.); \theta \in \Theta)$ of fuzzy numbers, denoted by $\pi^*(\theta|D^*)$, i.e.

$$\pi^*(\theta|D^*) = (\psi_\theta(.); \theta \in \Theta) .$$

For exact data D this definition reduces to the standard a-posteriori distribution $\pi(\theta|D)$.

7. HPD*-Regions for Parameters

In case of a parametric life time model $T \sim f(.|\theta)$ with parameter space Θ, a generalization of the concept of HPD-regions in Bayesian inference can be constructed for fuzzy data using the concept of fuzzy subsets of the parameter space Θ. For fuzzy data D^* and corresponding combined fuzzy sample $\varphi(\underline{x})$ and confidence level $1-\alpha$ a fuzzy subset of Θ with characterizing function $\chi(.)$ is defined by

$$\chi(\theta) = \sup\{\varphi(\underline{x}) : \int_{C(\theta,\underline{x})} \pi(\theta'|\underline{x})\,d\theta' \leq 1-\alpha\}$$

where the subset $C(\theta,\underline{x}) \subseteq \Theta$ is defined by

$$C(\theta,\underline{x}) = \{\theta' \in \Theta : \pi(\theta'|\underline{x}) \geq \pi(\theta|\underline{x})\}$$

for $\underline{x} \in [0,\infty)^n$.

The fuzzy subset θ^* defined by the characterizing function $\chi(.)$ is the so called HPD*-region for the parameter θ for confidence level $1-\alpha$. In figure 4 examples of HPD*-regions for the parameter of an exponential distribution are given.

Figure 4: Fuzzy HPD*-Regions

In case of exact life time data this definition reduces to the usual definition of HPD-regions.

8. Predictive Reliability Function

In Bayesian inference for parametric life time model $T \sim f(.|\theta)$, $\theta \in \Theta$ and a-priori distribution $\pi(\theta)$ of the parameter the reliability function conditional on the observation of exact Data D.

$$R(t|D) = Pr\{T > t|D\} \quad \text{for all} \quad t \geq 0,$$

also called predictive reliability function, is given by

$$R(t|D) = \int_t^\infty \int_\Theta f(x|\theta)\pi(\theta|D)d\theta\, dx \ . \qquad (8.1)$$

For fuzzy life time data $D^* = (x_1^*, \ldots, x_n^*)$ with corresponding characterizing functions $\varphi_1(.), \ldots, \varphi_n(.)$ the combined fuzzy sample $\varphi(\underline{x})$ can be used in order to generalize the predictive reliability function. For fixed t and $\underline{x} \in [0,\infty)^n$ using equation (8.1) the value $R(t|\underline{x})$ of the predictive reliability function using the corresponding a-posteriori distribution $\pi(\theta|\underline{x})$ is calculated. By variation of $\underline{x}$ according to $\varphi(\underline{x})$ a fuzzy number $\tau_t(.)$ is obtained by

$$\tau_t(y) = \sup\{\varphi(\underline{x}) : R(t|\underline{x}) = y\} \ .$$

The family $(\tau_t(.);\ t \geq 0)$ of fuzzy numbers constitutes the fuzzy predictive reliability function

$$R^*(t|D^*) = (\tau_t(.);\ t \geq 0).$$

In case of exact data this definition reduces to the standard Bayesian predictive reliability function.

9. Conclusions

Reliability estimation for fuzzy data is important for extrapolations from accelerated life tests to life

time distributions under usual environmental conditions because the fuzziness of observations under high stress levels is multiplied by extrapolation to low stress levels (compare [6]).

Estimation of the reliability of systems using fuzzy life time data of components is possible similar to the analysis for exact life time data.

By practical reasons it is necessary to model the imprecision in life time data. This can be done by using fuzzy numbers in a way which is feasible with reasonable amount of calculations and takes care of the imprecision of real data.

References

[1] J.O. Berger: Statistical Decision Theory and Bayesian Analysis, 2nd edition, Springer-Verlag, New York, 1985.

[2] G.J. Klir and T.A. Folger: Fuzzy Sets, Uncertainty, and Information, Prentice Hall, Englewood Cliffs, N.J., 1988.

[3] R. Kruse and K. Meyer: Statistics with Vague Data, Reidel Publ., Dordrecht, 1987.

[4] S.K. Sinha: Reliability and Life Testing, Wiley Eastern, New Delhi, 1986.

[5] R. Viertl: Is it necessary to develop a Fuzzy Bayesian Inference? in. Probability and Bayesian Statistics, Plenum, New York, 1987.

[6] R. Viertl: Statistical Methods in Accelerated Life Testing, Vandenhoeck & Ruprecht, Göttingen, 1988.

[7] R. Viertl and H. Hule: On Bayes' Theorem for Fuzzy Data, submitted for publication, Research Report RIS-1988-6, Institut für Statistik und Wahrscheinlichkeitstheorie, Technische Universität Wien, 1988.

[8] R. Viertl: Modeling of Fuzzy Measurements in Reliability Estimation, in. V. Colombari (ed.): Reliability Data Collection and Use in Risk and Availability Assessment, Springer-Verlag, Berlin, 1989.

Figure 1: Fuzzy Number

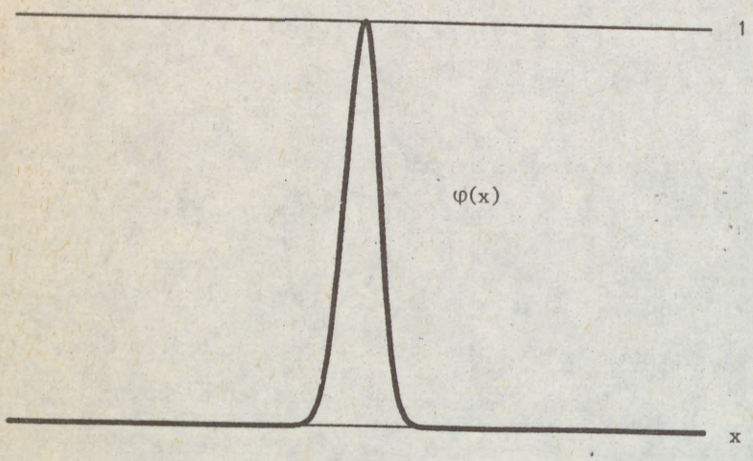


Figure 2: Fuzzy Empirical Reliability Function
for Fuzzy Life Time Data

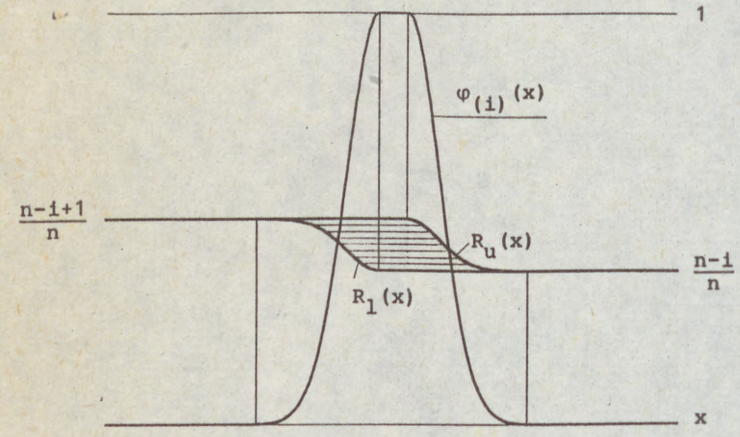


Figure 3: Fuzzy Empirical Reliability Function
for Interval Life Time Data

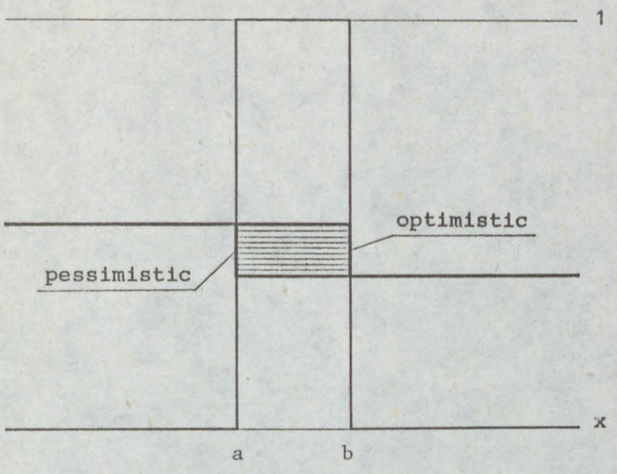
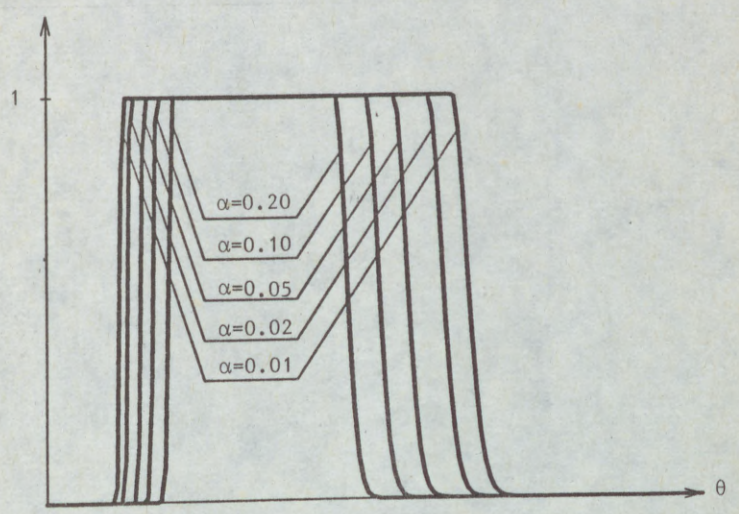


Figure 4: Fuzzy HPD*-Regions with Confidence Level 1-α



STATISTICAL QUALITY CONTROL - QUO VADIS?

Elart von COLLANI, Universität Würzburg

Germany

Statistical Quality Control - Quo Vadis?*)

Elart von Collani, Würzburg**)

1. Introduction

"Many thousands of dollars worth of product changes hands hourly, lots subjected to acceptance or rejection, depending on tests of samples, drawn from the lots. Examples of such plans are Military Standard 105D, and Dodge-Romig AOQL or Dodge-Romig LTPD. Such plans can only increase costs. If used for quality andit of final product as it goes out the door, they guarantee that some customers will get defective product. The day of such plans is finished. American industry can not afford the losses that they couse.
Incredibly, courses and books in statistical methods still devote time and pages to acceptance sampling."

This citation is taken from W.E. Deming's latest fundamental book "Out of the Crisis", which was published 1986. Since then many books on Statistical Quality Control have been published, but almost none of them took into consideration the change of industrial environments. Moreover since 1986 there have been taken place many meetings of the "International Standardization Organisation" discussing things related to statistical quality control, but evidently omitting to discuss the fundamental claims on acceptance sampling brought forward by Deming.

2. Acceptance Sampling by Variables

In some way the crisis of statistical as well as industrial

*) Research supported by the Deutsche Forschungsgemeinschaft (DFG).
**) Prof. Dr. Elart von Collani, Lehrstuhl für Statistik der Universität Würzburg, Sanderring 2, D-8700 Würzburg, West Germany.

quality is even deeper than observed by Deming. I would like to illustrate this statement by briefly investigating the well-known international standard ISO 395 1, which is a slight modification of the US-NilStd 414. The standard ISO 395 1 contains so-called sampling plans for variables which are looked upon as a cost-efficient alternative to attribute sampling.

The principle prerequisite for variables sampling is that the quality characteristic is measured on a continuous scale. The simplest case of a variables plan is a single sampling plan of type (n,c). In this case a random sample of size n is drawn from a lot submitted for inspection, and the quality characteristic of each item sampled is measured. If a suitable test statistic based on these measurements exceeds the acceptance number c, then the lot is rejected otherwise it is accepted.

About sampling by variables the following claim is stated unanimously:

> "Variables sampling provides more information than attributes sampling, and therefore the same protection is reached with partly considerably smaller sample size."

Here protection refers to the so-called consumer's risk and producer's risk respectively. These risks are defined as the probability of acceptance of a "bad" lot and the probability of rejection of a "good" lot respectively, where "bad" refers to a "large" fraction nonconforming p in the lot submitted for inspection, and "good" refers to a "small" fraction nonconforming p in the lot. In general variables sampling plans are determined in the following way: For a fixed attributes sampling plan with given risks one looks for a variables sampling plan with more or loss the same risks and minimum sample size. This problem is called "matching problem" and there are numerous papers dealing with it.
Calculation of the risks constitutes generally no problem for an attributes sampling plan. Let us look closer to this problem when a variables sampling plan is used. For the sake of simplicity we assume the simplest case, i.e. the so-called "$\bar{X}$-method" when a single (upper) specification limit U is involved and

the standard deviation is known. Then the procedure of applying a variables sampling plan (n,c) to a lot of size N is the following:

Select a random sample of size n out of the lot of size N. Compute the sample mean $\bar{x}$.
If $\bar{x} \leq U - c\sigma$ accept the lot, and
if $\bar{x} > U - c\sigma$ reject the lot.

In order to determine a suitable sampling plan (n,c) we have to calculate the risks. Let us concentrate on the producer's risk, i.e. the probability of rejecting a "good" lot. Clearly, a lot containing no item nonconforming at all (p=0) constitutes a good lot, and one would expect that such a lot is accepted with certainty. For proving it, the following quantity has to be calculated:

$P_a(0)$ = Prob (accepting a given lot, when using the variables sampling plan (n,c) | the lot contains no item nonconforming).

Of course, we have

$$0 \leq P_a(0) \leq 1. \tag{1}$$

Each lot of size N is given by an element of $\mathbb{R}^N$, i.e.:

$(x_1, x_2, \ldots, x_N) \in \mathbb{R}^N$.

Obviously the subset A represents all possible faultless lots (p=0):

$$A := \{(x_1, x_2, \ldots, x_N) \mid x_i \leq U, i=1,2,\ldots,N\}.$$

Let

$$A_1 := \{(x_1, x_2, \ldots, x_N) \mid U-c\sigma < x_i \leq U, i=1,2,\ldots,N\}$$

and

$$A_2 := \{(x_1, x_2, \ldots, x_N) \mid x_i \leq U-c\sigma, i=1,2,\ldots,N\}$$

then A_1 and A_2 are subsets of A, and therefore each element of A_1 and A_2 represents a lot with fraction nonconforming p=0.

And of course for any relevant situation we have
Prob $(A_1) > 0$, and
Prob $(A_2) > 0$.

Let us return to the probability $P_a(0)$ and assume that the lot $(x_1, x_2, \ldots, x_N)$ in question is an element of A_1, then obviously we obtain:

$$P_a^{(A_1)}(0) = 0.$$

Next consider the case that the faultless lot in question $(x_1, x_2, \ldots, x_N)$ is an element of A_2, then

$$P_a^{(A_2)}(0) = 1.$$

Hence we have obtained the result that the inequalities (1) are sharp in the sense that both the trivial lower as well as the trivial upper bound of $P_a(0)$ are adopted with positive probability. It may be concluded that $P_a(0)$ can adopt almost any value in $[0,1]$ solely depending on the particular realization of the random vector $(X_1, X_2, \ldots, X_N)$ representing a lot of size N. In other words using a variables sampling plan the decision on a faultless lot becomes a mere lottery, and, of course, this statement holds analogously for different values of the lot fraction nonconforming $p(\neq 0)$ too.

This example illustrates the situation of today's statistical and industrial quality control:
Variables sampling recommended in many textbooks on statistical quality control and by national and international organizations for industrial application does not at all constitute a more efficient alternative to attribute sampling, but has the following properties:

1) It may happen that a faultless lot is rejected with probability 1 even in the case of 100% inspection.
2) It may happen that a lot containing only items nonconforming is accepted with probability 1 even in the case of 100% inspection.

3) It may happen that the risks are monotonously increasing in the sample size n.

In conclusion traditional quality control as developed decades ago is not "repairable" and one should think of new ways and a new approach being more appropriate to modern industrial environments.

3. Economic Quality Control

The overall aim of traditional quality control is to develop sorting procedures, i.e. methods how to sort good items or lots from bad. The aim of sorting procedures is to improve the outgoing quality but, of course, they are not applied to improve the produced quality.

But today it is common knowledge that a company which relies on such sorting procedures cannot have economic success at least on a longterm basis. Producing bad quality means wasting expensive resources and leads almost necessarily to an economic failure.
In today's "New Economic Age" only those companies will operate successfully in the long run which manage to produce high quality products meeting all relevant consumer's demands. Therefore the production process should be in the center of scientific investigations on the one hand and industrial quality control procedures on the other hand.

The history of any production process can be roughly divided into two parts:
1) the design and installation phase,
and
2) the operational phase, where mass production is performed.

The aim during the first phase is to develop a production process or system which has the following properties:

- the quality produced is acceptable
- the process variability is small
- the process is insensitive against random changes of the process environment.

As soon as the production process has reached such a desirable state, the operational phase can be started. During this final phase management is responsible for it and in particular for conserving the once achieved satisfactory state throughout the lifetime of the process.
Therefore statistical quality control should provide management with efficient policies to meet the task to maintain the high quality level of the process.
In order to be efficient a policy has to take into account the technical/statistical process parameters as well as the economic environment of the process; in order to meet its purpose, it has to answer the question when to perform which control and/or maintenance action. Therefore "Economic Quality Control" can be describe as an optimal control and maintenance theory for industrial production processes.

Any optimal control and maintenance policy has to be determined in three steps:

Step 1: Specification of the underlying mathematical model, i.e. process model, control and maintenance model, economic model.

Step 2: Derivation of a suitable (economic) objective function for optimization.

Step 3: Development of a (simple) solution algorithm for the optimization problem.

In general the output of a production process can be divided in more or less discrete units. Let the i-th unit be represented by a random variable X_i, then it is clear that a production process can be described mathematically by a discrete stochastic process $\{X_i\}_{i=1,2,\ldots}$.
Each item produced X_i is judged by a reward (profit or loss),

therefore $\{X_i\}_{i=1,2,\ldots}$ is a discrete stochastic reward process. Hence in order to define a mathematical model, we have to specify the characteristic properties of a discrete stochastic reward process, which very often can be looked upon as a renewal reward process.

A production process is part of an economic enterprise aiming in deriving as much profit as possible on a longterm basis. Let m be the number of items produced and P(m) the profit gained from these items, then the average profit per item is given by P(m)/m. Hence it is reasonable to select the following limit as objective function for optimization with respect to the control and maintenance policies:

$$\pi^* = \lim_{m\to\infty} \frac{P(m)}{m}$$

which can be interpreted as longterm average profit per item.

Let

$\Gamma = \{\gamma\}$

be the set of admissible control and maintenance policies, then π^*, of course, depends on the policy γ selected for use.

Definition: Let $\gamma^* \in \Gamma$, then γ^* is called optimal with respect to (π^*, Γ) iff

$\pi^*(\gamma^*) \geq \pi^*(\gamma)$ for any $\gamma \in \Gamma$.

The following simple example serves to illustrate the proceeding on the one hand and the difference to traditional quality control on the other hand.

4. Simple Example for Illustration (Maintaining the Process Mean)

4.1 The Process Model

$\{X_i\}_{i=1,2,\ldots}$ is a sequence of independent, normally random variables with constant (and known) variance σ^2.

The process $\{X_i\}_{i=1,2,\ldots}$ may adopt two states:

State I: $E[X_i] = \mu_0$
where μ_0 is called target value.

State II: $E[X_i] = \mu_0 \pm \delta\sigma$
where δ is assumed to be known and is called the shift parameter.

Transition behaviour:

Let

τ = time of operation in State I after start until a transition to State II.

It is assumed that τ has an exponential distribution with parameter $\lambda(>0)$:

$$P(\tau \leq t) = \begin{cases} 0 & t < 0 \\ 1-e^{-t} & t \geq 0 \end{cases}$$

with $E[\tau] = 1/\lambda$.

A transition from State II back to State I is possible only by means of a corrective action.

4.2 The Set of Admissible Actions

For this simple example we consider three actions to be performed:

1) sampling, revealing the actual state of the process with probability smaller than 1.

2) inspection, revealing the actual state of the process with probability 1.

3) renewal, restoring with probability 1 State I.

In view of the exponential transition model only periodic policies make sense, hence we define

$\Gamma_p := \{(h,n,c): h>0,\ 0 \leq n \leq h\nu\ ,\ c \geq 0\}$

where ν is the production speed.

A policy $(h,n,c) \in \Gamma_p$ works as follows:

Every h hours of operation take a sample of size n and calculate the test statistic $T := \left| \frac{\bar{x}-\mu_0}{\sigma} \sqrt{n} \right|$.

If

$T < c$ then the process continues to operate,

if

$T \geq c$ then an alarm is released, an inspection and if necessary a renewal are performed.

By elementary calculations we obtain the following error probabilities in the case $1 \leq n \leq h\nu$ and $c > 0$:

α = Prob(false alarm) = $2\Phi(-c)$

β = Prob(not detecting State II) = $\Phi(c-\delta\sqrt{n}) - \Phi(-c-\delta\sqrt{n})$

where Φ denotes the distribution function of the standardized normal distribution.

For $n = c = 0$ it follows:

$\alpha = 1$ and $\beta = 0$.

Therefore the case $n = c = 0$ can be looked upon as a routine inspection policy, i.e. (h,0,0) means that every h hours of operation an inspection and if necessary a renewal are performed.

Further assumptions for simplification:

- the time for drawing and evaluating the sample is negligible small,
- the probability of a transition during sampling is negligible small.

4.3 Economic Model

The economic model must describe the process on the one hand and the actions performed on the other hand, therefore we have two types of economic parameters:

1) Economic parameters describing the process $\{X_i\}_{i=1,2,\ldots}$:

g_I^* = average profit per item produced while operating in State I.

g_{II}^* = average profit per item produced while operating in State II.

Obviously we have $g_I^* > g_{II}^*$.

2) Economic parameters describing the policy (h,n,c):

a_n^* = cost of drawing and evaluating a sample of size n.

e^* = average cost of an (erroneous) inspection following a false alarm.

b^* = average benefit per renewal, i.e. additional profit gained for operating in State I after a renewal reduced by the cost of the final inspection and renewal.

We assume $a^*>0$, $e^*>0$ and $b^*>0$.

4.4 The Objective Function

The start of the process and each renewal are socalled generative points. The time between two successive generative points is called renewal cycle. From the assumptions it immediately follows that different renewal cycles are stochasticalle equivalent.

Each of the following random variables refers to a renewal cycle:

P = profit,

N = number of items produced,

A_I = number of sampling actions while operating in State I,

A_{II} = number of sampling actions while operating in State II,

A_F = number of false alarms.

From renewal theory we have with probability 1:

$$\pi^*(h,n,c) = \frac{E[P]}{E[N]} ,$$

therefore it remains to determine the expectations of P and N.

Using the above defined quantities, we obtain:

$$E[N] = E[A_I+A_{II}]\, h\nu$$

and

$$E[P] = E[A_I+A_{II}]h\nu\, g_{II}^* + b^* - E[A_F]e^* - E[A_I+A_{II}]a^*n.$$

Hence

$$\pi^*(h,n,c) = \frac{b^* - E[A_F]e^* - E[A_I+A_{II}]a^*n + E[A_I+A_{II}]h\nu\, g_{II}^*}{E[A_I+A_{II}]\, h\nu} =$$

$$= \frac{1}{h\nu}\left\{\frac{b^* - E[A_F]e^*}{E[A_I+A_{II}]} - a^*n\right\} + g^*_{II}.$$

Obviously the following transformation of π^* is equivalent with respect to the optimization:

$$\frac{\pi^*(h,n,c)-g^*_{II}}{e^*}\cdot\nu = \frac{1}{h}\left\{\frac{b - E[A_F]}{E[A_I+A_{II}]} - an\right\}$$

with $b = b^*/e^*$ and $a = a^*/e^*$.

Up to now we have not utilized neither the special distributional assumptions nor the special sampling procedure selected. Note the generality and comparative simplicity of the objective function. These two almost characteristic features of this approach made it possible to develop a rather generally applicable approximation technique to obtain a simple solution algorithm.

From the model assumptions we get by elementary calculations:

$$E[A_I] = \frac{1}{e^{\lambda h}-1},$$

$$E[A_{II}] = \frac{1}{1-\beta},$$

$$E[A_F] = \frac{\alpha}{e^{\lambda h}-1}$$

and therefore:

$$\pi^*(h,n,c) = \frac{1}{h\nu}\left\{\frac{b^*(e^{\lambda h}-1)-\alpha e^*}{e^{\lambda h}-\beta}(1-\beta) - a^*n\right\} + g^*_{II}$$

or the socalled standardized objective function:

$$\pi(x,n,c) := \frac{\pi^*(\frac{x}{\lambda},n,c)-g^*_{II}}{e^*}\cdot\frac{\nu}{\lambda} =$$

$$= \frac{1}{x}\left\{\frac{b(e^x-1)-\alpha}{e^x-\beta}(1-\beta) - an\right\}$$

with $\alpha = 2\Phi(-c)$ and $\beta = \Phi(c-\delta\sqrt{n})-\Phi(-c-\delta\sqrt{n})$.

Obviously π is equivalent to π^* with respect to the optimization, and there are only three input parameters entering explicitly π: a, b and δ.

4.5 The Solution Algorithm

A further simplification of the problem is obtained by the following transformation:

$$y := \delta\sqrt{n} \iff n = \left(\frac{y}{\delta}\right)^2$$

$$z := c$$

$$a_0 := \frac{a}{\delta^2}$$

and

$$\tilde{\pi}(x,y,z) = \pi\left(x,\left(\frac{y}{\delta}\right)^2,z\right) = \frac{1}{x}\left\{\frac{b(e^x-1)-\alpha}{e^x-\beta}(1-\beta) - an\right\}$$

with $\alpha = 2\Phi(-z)$ and $\beta = \Phi(z-y) - \Phi(-z-y)$.

This leads to a first approximate solution for our problem:

> Let (x^*,y^*,z^*) with $x^*>0$, $y^*\geq 0$, $z^*\geq 0$ be an optimal solution with respect to $\tilde{\pi}$, then
> $\hat{n}_1^* = x^*/\lambda$,
> $\hat{n}_{i1}^* =$ nearest integer to $\left(\frac{y_1^*}{\delta}\right)^2$,
> $\hat{c}_1^* = z_1^*$,
> is an approximate optimal policy with respect to (π^*,r_p).

In order to determine a relative maximum of $\tilde{\pi}$ the following necessary conditions have to be met:

$$\frac{\partial\tilde{\pi}}{\partial x} \stackrel{!}{=} 0, \tag{2}$$

$$\frac{\partial\tilde{\pi}}{\partial y} \stackrel{!}{=} 0, \tag{3}$$

$$\frac{\partial\tilde{\pi}}{\partial z} \stackrel{!}{=} 0. \tag{4}$$

From (1) the following relation for the optimal x* is obtained:

$$\frac{b(e^x-1)^2}{e^x-\beta} = \frac{2(a_0y^2+\alpha)}{1+\beta} + O(x). \qquad (5)$$

The optimal x* gives the optimal control interval measured in time-units $1/\lambda$, i.e. the expected lifetime of State I. Therefore x* is a small quantity.
Substituting (5) into (3) and (4) and neglecting all terms $O(x)$ (being small), we finally obtain the following equations for determining approximately the optimal y* and z*:

$$a_0y + \frac{a_0y^2+\alpha}{1-\beta^2}\beta_y = 0$$

$$\alpha_z + \frac{2(a_0y^2+\alpha)}{1-\beta^2}\beta_z = 0$$

with

$$\alpha_z = \frac{\partial\alpha}{\partial z}\ ,\ \beta_z = \frac{\partial\beta}{\partial z} \text{ and } \beta_y = \frac{\partial\beta}{\partial y}\ .$$

These equations are called key equations.

The "key equations" together with (5) lead to the following simple solution algorithm:

Step 1: Take from Figure 1 for given value of $a_0 = \frac{b^*}{e^*\delta^2}$ the two quantities $\hat{y}^*$ and $\hat{z}^*$.

Step 2: Compute the quantity

$$\hat{x}^* = \ln\left\{1+\frac{a_0\hat{y}^*+\hat{\alpha}^*}{b(1+\beta^*)}+\sqrt{\left(\frac{a_0\hat{y}^*+\hat{\alpha}^*}{b(1+\beta^*)}\right)^2+\frac{2(a_0\hat{y}^*+\hat{\alpha}^*)}{b(1+\beta^*)}(1-\beta^*)}\right\},$$

where $\hat{\alpha}^* = 2\Phi(-\hat{z}^*)$ and $\hat{\beta}^* = \Phi(\hat{z}^*-\hat{y}^*)-\Phi(-\hat{z}^*-\hat{y}^*)$.

Step 3: The approximate optimal control and maintenance-policy with respect to $(\varkappa^*,\Gamma_p)$ is given by:
$\hat{h}^* = \frac{\hat{x}^*}{\lambda}$
$\hat{n}^*$ = nearest integer to $(\frac{\hat{y}^*}{\delta})^2$
$\hat{c}^* = \hat{z}^*$.

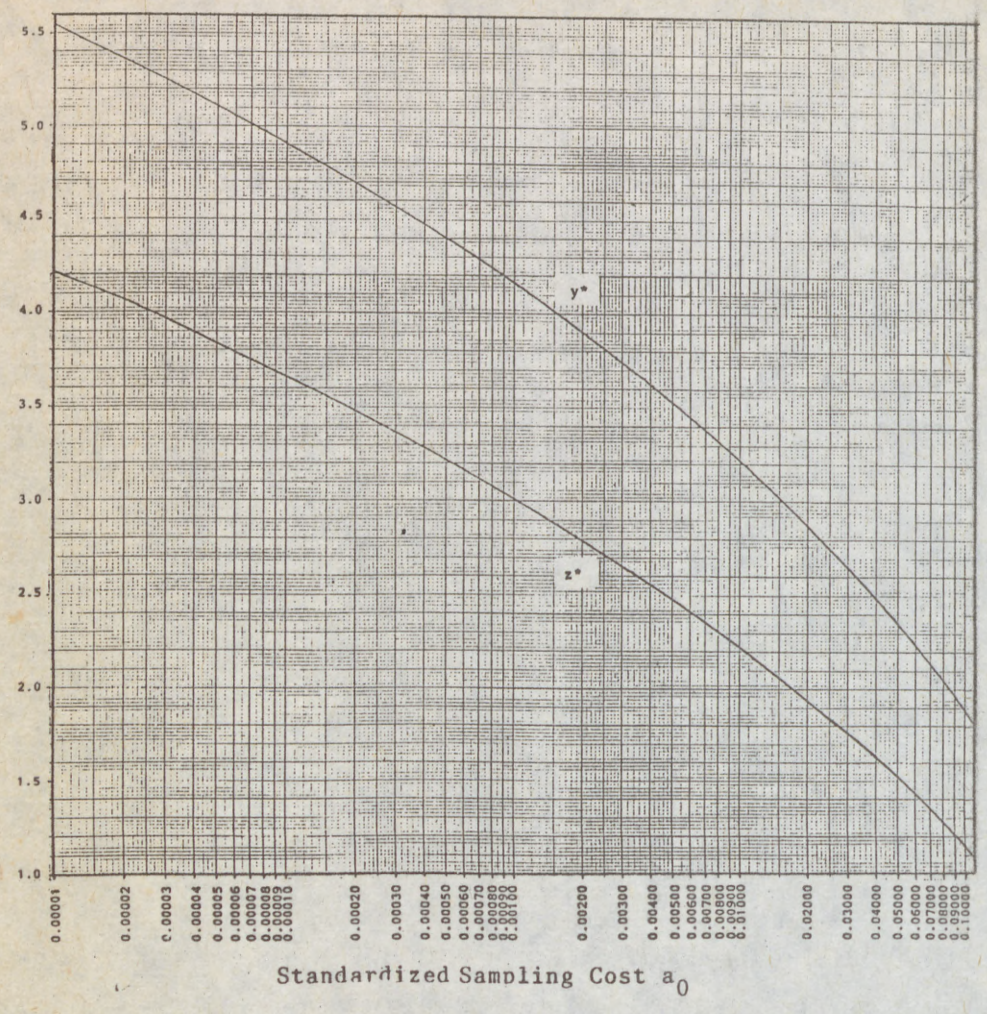


FIGURE 1

Remark: The approximation technique outlined here is applicable in a great number of case, for instance in the multivariate case, in the case of general lifetime distribution and for arbitrary sampling procedures.

5. Conclusions

Many of the traditional quality control methods are applied just "pro forma", merely conforming to the requirements of a contract or because everybody is using them.
In order to save money and as a consequence improve the competiveness the comparnes should give up this practice and try to apply adapted policies assuring economic success to the widest extent.

References:

Collani, E.v. (1989). The Economic Design of Control Charts. Teubner Verlag, Stuttgart.

Collani, E.v. (1990). Wirtschaftliche Qualitätskontrolle - eine Übersicht über einige neue Ergebnisse. OR Spektrum 12, 1-23.

Collani, E.v. (1990). A Note on Acceptance Sampling for Variables. Metrika (to appear).

Deming, W.E. (1986). Out of the Crisis. MIT, Cambridge, USA.

Dodge, H.F. & H.C. Romig (1959). Sampling Inspection Tables. Single and Double Sampling. 2nd ed. Wiley, New York.

FAILURE MODE EFFECTS AND CRITICALITY ANALYSIS; RECENT IMPROVEMENT FOR DESIGN BASED ON A COMMON DATA BASE FOR A LARGE PROJECT

P. R. LECLERCQ, Matra D3E

France

Advanced Studies on Reliability Engineering
(A.S.R.E.)
at the Technical University of Budapest (T.U.B.)

Failure Mode Effects and Criticality Analysis ;

Recent Improvement for Design based on a Common Data Base for a Large Project

P.R. LECLERCQ
MATRA D3E- Dependability Manager
37, avenue Louis Bréguet - 78146 Vélizy Villacoublay Cedex - France

SUMMARY

ABSTRACT

We propose, successively, to present on FMECA, briefly, what it is, what it is not.
Based on these considerations the problems accounted during performance this analysis we present some improvements developed in order to be more efficient.
The most significative is the availability of data base that all the partners a project may access.

KEY WORDS: FMECA, Methodology, Software tool, Data base.

1. INTRODUCTION

The FMECA is a major method in order to obtain the Dependability(*) criterias a product (equipment, system, ...) allowing to point out risks, with the objective to reduce them later by appropriate measures.

The FMECA force to perform rigourous analysis which have to be introduced at the earliest phase of a project when corrective actions are possible and the corresponding cost is low.

(*) Dependability is used as equivalent to RAMS : Reliability, Availability, Maintenability, Safety.

2. RECALL ON FMECA

2.1. What is a FMECA ?

The FMECA is an inductive method. That means from a failure mode the analysis consists to look at the potential effect(s) at different integration levels a product (printed circuits, equipments, sub-systems, finally system).

In addition the reliability expert is looking for the criticity of the effect on the mission either the impact may an unavailability or a safety event.

Finally, he tries to determine if failure mode may be observed by the different tests performed on the material at different integration levels or by telemetry in operation. Then it is possible to compair with tests or telemetry checklists and to perform an estimation of the test coverage rates.

In addition, it is possible to determine for each failure mode if it exists in operation a compensation measure designed for that purpose (ie : redundancies, acceptable degraded missions, ...).

If the reliability expert uses FMECA it constitutes an improved tool for designing, which when used sufficiently early in a project with all the partners involved in the design is able to avoid a lot of discrepencies during the project development.

2.2. What is not the FMECA ?

A lot of people suppose that FMECA can be an help for the study of effect of multiple failure modes with occurrence in the same period. Generally it is not true except in particular cases like redundancies. In fact, by the princip of the method, which is inductive, the analysis must be conducted failure mode by failure mode in an exhaustive fashion. A study at an upper level (combinaison of failure modes) will conduct to a very large increase of the cases to be studied. This is done on potential critical areas and cannot be qualified as exhaustive.

For that last purpose it is necessary the customer, the main contractor, after preliminary studies, defines the effects against he intends to be protected. This may be introduced in a particular paragraph of the specifications.

In order to allow the reliability analysis more efficient it is necessary to associate a deductive method like FTA (Fault Tree Analysis) to FMECA.

Finally, we have to keep in maind that we do study only the effects of failure modes as described in related catalogs. These catalogs suppose that technologies are correctly implemented. In fact we cannot consider that approach as exhaustive. There is one the interest of clue lists, which are catalogs of configuration of past failures it is necessary to avoid the occurrence in the new designs.

3. APPLICATION AREA

3.1. Hardware application

FMECA is applicable on all assemblies on which the failure of one its assemblies conduct to do not meet Dependability objectives.

More precisely, the application may be :

- a product : FMECA product oriented
- a process : FMECA process oriented based on production/process operations
- a production tool : FMECA related to the design the operation of production tools.

The methodology have the same basis for these different application cases, only differ the analysed events and the production the FMECA.

Remark : The application domains corresponding to these three types of FMECA may have some common coverage.

3.2. Software application

On software application the processing is similar, nevertheless expert on this area speak of SEEA (Software Error Effect Analysis).

In order to limit this presentation we will not describe any more this particular analysis.

4. REFERENCE DOCUMENTS

The approach, how to perform a FMECA is defined in numerous specifications.

We can list as exemples :

- MIL - STD 1629A
- ESA PSS 01-303
- CNES SR 1-40
- CEI 812
- AFNOR 60-510
- ..., and other documents issued by other agencies or main contractors.

On a common basis these documents are related mainly to FMECA with product application.

We do not detail them, because it is not in the main topic of this paper.

5. PROBLEMS WHICH OCCUR DURING PERFORMANCE A FMECA

As FMECA are generally performed, they introduce some defects we intend to describe in the following items :

- They are time consuming analysis which results seem available too late in the design in order to be applicable. Nevertheless we must note that before the delivery to the customer the data package including the formal analysis, a lot of work has been performed directly with the designer which has conducted to cancel a maximum of the potential critical points.
- To get a quantitative FMECA in accordance with reliability prediction is a tedious task.
- The results of these analysis are presented through a thick data package on which it is difficult to perform an operation (search, ...) and more to update when the product is modified.
- The contained informations in a FMECA are difficult to complete at upper level. One solution is to use synthesis which are performed on simple basis but time consuming to be completed.
- Finally there is no guarantee of exhaustivity of the analysis. There are some risks of parts or failure modes omission and there are no mean to verify application of design rules.

6. RECENT IMPROVEMENTS FOR DESIGNING A LARGE PROJECT

6.1. Objectives definition

The problems listed previously have conducted to introduce some improvements.

These improvements are related to :

- objectives definitions
- organisational dispositions
- data base
- support tool.

All of these dispositions have for goal to get more efficiency on a project by usage of a data base, result of the FMECA, available by all the participants to the product development in order to get the just dependability level.

We will develop successively these different aspects.

6.1.1. Introduction

Even if from outside all the FMECA seems to be the same in fact for efficiency purpose the expert detailed his analysis in relation with project environment constraints. So different critirias must be considered before starting the analysis. These criterias must be defined in the specification related to the designed product. Two major criterias have to specified :

- FMECA purpose
- FMECA analysis level.

6.1.2. FMECA purpose

It is necessary to define what is the type of discrepancies we wish to discover. We list after that some of them related to :

- Reliability :
  - . single point failure ; points which failure induced mission interrupt or does not allow the usage of a redundancy ;
  - . failure propagation risk,
  - . degraded operational performance studies.
- Maintenability :
  - . ability to observe the failure through their effects,
  - . isolation of failed assemblies.

- Availability :

  . test efficiency determination,

  . mission interruption duration.

- Safety :

  . fail operational / fail safe criteria verification.

6.1.3. FMECA analysis level

A FMECA may be performed at different assembly levels ; the failure modes (FM) and effect(s) (EF) are then of different nature as hereafter :

- Macro-parts (ie : hybrids)
  FM micro-parts = = > EF functional bloc = = > EF macro

- Printed board
  FM parts = = > EF functional bloc = = > EF printed board

- Equipment
  FM functional bloc = = > EF equipment

- Sub-system
  FM equipment = = > EF sub-system

So the process is iterative and equivalent at each level.

6.2. Organisational dispositions

The FMECA is generally performed by the reliability expert. In fact it may be established by the designer at three conditions :

- he has a necessary experience on that methodology

- he is able to have a sufficient independant judgement on his own design

- he has motivation to perform that task in order it will be effectively implemented.

So in practice the share of tasks is as :

- Reliability expert : research the potential problems and guarantees on the methodology application and on the data.

- Designer : definition of the modification.

So the reliability expert must be also an expert of the studied domain (ie : electronic, mechanic, ...).

In any case the FMECA must be conducted in close collaboration between all partners involved in the development process. Figure 6.2. presents the most usefull partners and how they use the results of the FMECA.

Before FMECA performance it is necessary :

- to know the development plan the assembly in order the results may be applied after review,
- to get the contractual requirements,
- to define with the designer additional needs,
- to procure drafts of the schematic and definition report,
- to define blocs which will be the basis of the analysis.

6.3. Data base purpose

The analysis may be conducted by answer to a simple question :

WHAT HAPPEN IF a failure occurs, ... ?

IF no EFFECT THEN what is the purpose of the analysed assembly ?, ...

IF effect THEN description of the effect itself, observable symptoms, compensation methods, ...

Even if the analysis is based on a simple principle, rapidely it becames a very large sum of informations. So it is necessary to record the results in a data base who gather :

- the basic analysis,
- the synthesis which summarizes by effect all the causes.

By that mean all the informations may be available to the different partners a project after appropriate treatment.

6.4. Presentation the AMDEC* software tool

* A.M.D.E.C. "Analyse des Modes de Défaillance, de leurs Effets et de leur Criticité" (in french, équivalent to FMECA).

6.4.1. Introduction

For all these reasons the use of a software tool had appeared as a valuable assistance to solve all these problems. So the Dependability Department at MATRA Velizy had decided to develop a simple performance tool, with friendly user interfaces but with powerfull facilities when needed. Some of its main functions will be presented during the course.

6.4.2. General objectives

This tool has for objective to be at the interfaces between the different partners in the design of a product, like designers, ..., interested by search and analysis :

- single point failure (SPF)
- operational degraded modes
- failure propagation
- failures testability and observability.

For that purpose the tool is organized in order to establish a reference product data base easily updated.

The reliability expert by his reliability knowledge is the preferential user of that tool during analysis phase. There, it is necessary that one of the major goals of the designer can be related to that analysis, who finally has to take into account and is responsible for that task.

At project milestones the date base is then certified and take place in the project data package to be transmitted to the upper integration level or to the customer.

Finally the data base is an input for test lists and operational manual.

So this data base cannot be a static package but dynamic one all along the project and must be easy to be updated by the different partners at the successives phases.

6.4.3. Resumed specification

- direct input with the keyboard at the different integration levels (no more paper, only for some time for contractual formalism)
- facilities for input and update
- no particular software knowledge
- sort on multiple criteria
- synthesis at level Ni and preparation the input for analysis at level Ni + 1
- search of any information
- deductive analysis effects - failure causes
- link with parts nomenclature and failure modes
- link with reliability predictions (failure rates)
- on request issues (software file or paper)

6.4.4. Tool description

The tool is organized following the princips of the structured programming. To process the tool, the user go through menus that allow access to the modules that perform the needed functions.

The user is continualy driven in his work, he can use help utilities and predifined tables he can update for his own purpose.

This tool is an application developed on the basis the dBaseIII + and include around 4000 instructions in macro-language.

AMDEC may be implemented as it is on all PC's or compatible, nevertheless the minimum configuration for an efficient usage is :

- RAM memory : 640 K
- Clock : 8 MHz
- Hard disk drive.

AMDEC is available in two versions :

- The first one caracterized as mono-level can be used at every integration level (board, equipment, sub-system, system). That version is particularly adapted when the failures modes are at parts level. With this tool we may perform basic analysis and the corresponding analysis.

- the second version is referenced multi-level. In addition to the previous facilities, this version offer facilities to chain analysis at different hardware integration levels. By this tool we can after an analysis at one level generate automaticaly the failure modes at upper level and the analysis consists only on completion the effect and related informations.

  In addition, the tool allow to search on a marked effect we can trace all the possible causes. This is particularly interesting for designing test, there FMECA is an important input.

7. CONCLUSION

We have presented some improvements related to management, performance a FMECA. All of them conduct to get more efficiency by considering and establishing the result of the FMECA in a data base which is available by all the partners a project. This data base is available through a software tool developped for that purpose.

In addition AMDEC tool allow to :

- reduce analysis cost

- increase efficiency and quality of FMECA

- assure that all the parts and associated failure modes have been analysed

- make easier the use for the result the analysis by real time extraction an information in the data base

- perform synthesis, inputs for exemple for operation manual or for testability studies.

AMDEC®

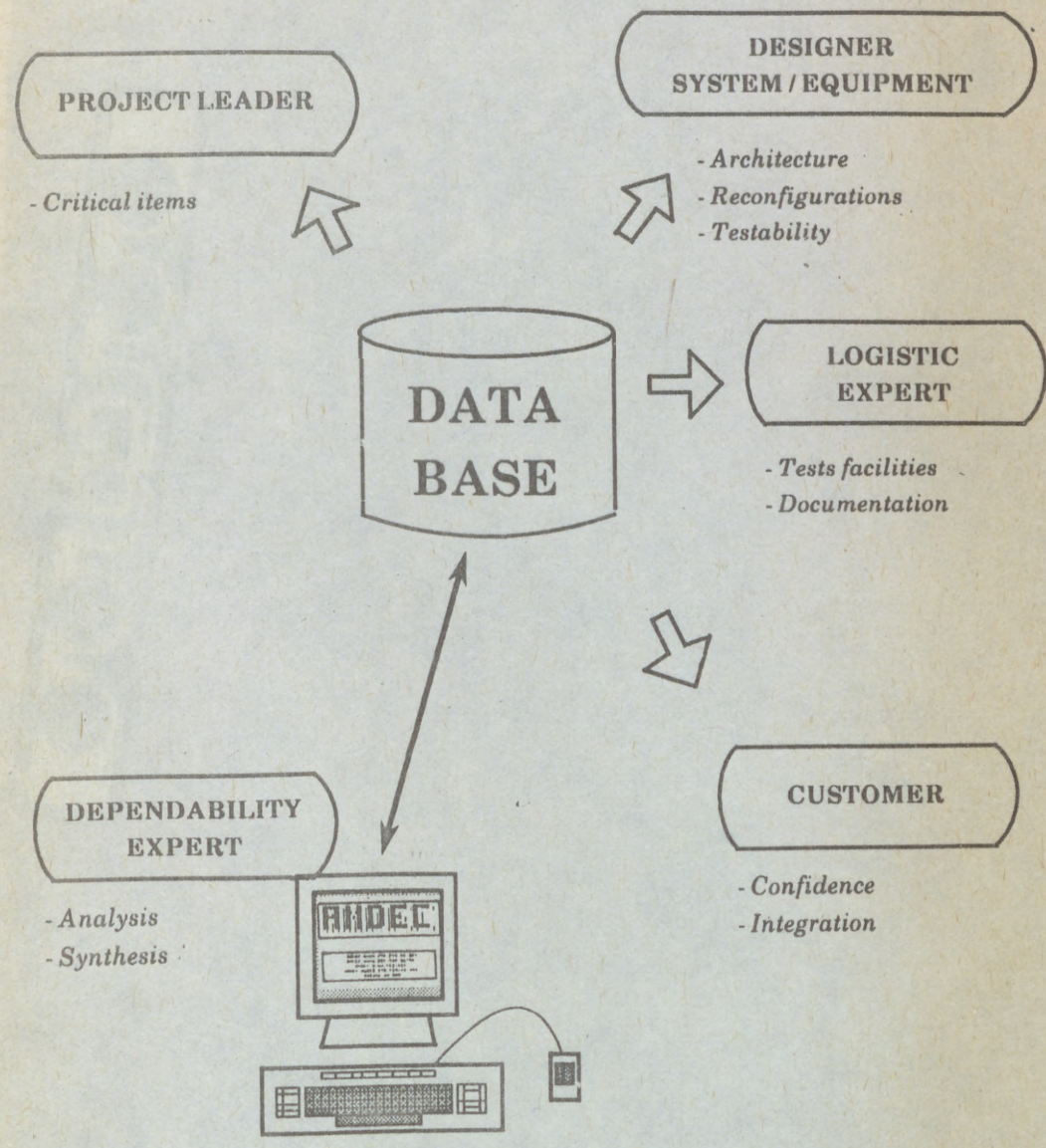


Figure 6.2.