Magnus Lundin, SP
Lars-Åke Johansson, CTH
Jan Jacobson, SP
Hanna Larsson,HiSafe Development AB

Safety of Distributed Machine Control Systems,
Validation Methods

SP Report 1998:24

Physics & Electrotechnics

This issue of the report does not contain descriptions of all the validation methods.

Please contact SP, Ms. Lisbeth Pilgard (Phone +46 33 16 53 84,

Email lisbeth.pilgard@sp.se) to order the complete printed report.

Abstract

Safety of Distributed Machine Control Systems, Validation Methods

Distributed control systems are being used more and more in machines. With distributed technology new types of errors are introduced. A pre-study has earlier resulted in identification of seven different types of errors which are considered to be unique for a distributed system.

This report describes methods for validating the safety of a distributed control system. The aim with this validation method is to cover all the error types identified in the earlier pre-study. The methods are developed to suit several different communication protocols, and not only CAN, even if the CAN protocol has been studied in parallel with this work.

Key words: Safety of machinery, distributed control systems, safety validation, assessment method

SP Sveriges Provnings- och Forskningsinstitut
SP-Rapport 1998:24
ISBN 91-7848-730-7
ISSN 0248-5172
Borås 1998

SP Swedish National Testing and Research Institute
SP Report 1998:24

Postal address:
Box 857, S-501 15 Borås
Sweden
Telephone +46 33 16 50 00
Telefax +46 33 13 55 02

Contents

Preface

The traditional way of controlling machines has been to use a centralised computer. All sensors, actuators, displays etc. are connected to the central computer. Today more and more of the machines are built up with several different computers which are connected together. The different computers (nodes) can exchange information in real-time in order to be able to control the machine. Safety critical applications such as excavators and packaging machines are today built up with distributed technology.

The distributed technology has a lot of advantages such as modularity and flexibility, but new risks have been introduced as well. A study of these risks which are unique in a distributed system, were done in 1996. This work has been documented in SP Report 1996:23.

The objective of the research work described in this report is to develop validation methods for safety in distributed machine control systems.

The project team has members from the following companies:

- TetraPak
- Scania
- VOAC Hydraulics Division
- KVASER
- Chalmers University of Technology
- IVF, Swedish Institute of Production Engineering Research
- SP Swedish National Testing Research Institute
- HiSafe Development

Thanks are directed to all members for their efforts, hard work and useful contribution to the project.

Thanks are also directed to the reference group who supervised the pre-study and the start of this work. The reference group has members from:

- Swedish Metal Workers Union (Metallindustriarbetarförbundet)
- Swedish National Board of Occupational Safety and Health (Arbetarskyddsstyrelsen)
- Association of Swedish Engineering Industries (Sveriges Verkstadsindustrier)
- ELLÅÅ Ingenjörsbyrå
- IVF, Swedish Institute of Production Engineering Research
- SP Swedish National Research and Testing Institute

The work has been financially supported by the Swedish Council for Work Life Research (Rådet för arbetslivsforskning)

Summary

A set of methods and techniques for validation of functional safety of distributed control systems is given.This report’s focus is on safety-related applications for machinery, both permanently installed and mobile equipment.

Most of the existing validation methods for programmable electronic systems do not sufficiently cover the faults that may be introduced by the use of communication buses. This new set of validation tools will supplement the ”established” methods. The conclusion on functional safety will be based both on the result of the general methods, and on the result of the validation methods of this report. Both the general risks associated with programmable electronic systems, and the specific risks associated with the use of communication buses will then be covered.

The validation methods can be grouped in three parts; validation of specification, analysis methods and test methods. These three parts include techniques to validate node errors, bus errors, timing errors, data consistency errors, initialisation/restart errors, babbling idiot errors and configuration errors.

It has been demonstrated that it is possible to select methods to check for faults in distributed control systems. As always in programmable electronic systems, it will not be possible to guarantee a fault-free design. The degree to which the validation is carried out will have to be specified for each system. A minimum requirement for many systems will be to address all the unique fault types of a distributed control system by at least one validation method.

There is a need to be able to validate functional safety in distributed control systems. This need will grow even stronger as the number of applications continue to increase. Future research work should include more use of the validation methods on different applications.

1 Introduction

The traditional way of building computer based machine control systems has been to use a central computer. All sensors, actuators, displays etc. have been connected to one central point where the control system is located. Until now it has been easy to identify where the intelligence is located, and requirements for centralised systems can be well defined.

Machine based systems of today are often built up by several different computers which are connected together via a communication bus. Inputs and outputs have been distributed out to positions where the sensors and actuators are located. In many cases also the intelligence has been distributed to the machine parts it is intended to control.

An example of a distributed system can be the hydraulic system in a mobile machine. The hydraulic components are getting more intelligent with processor power, and the control on system level is made by sending information from/to the different modules. Another example can be the reading of safety switches on a machine. All safety switches indicating over travel, opening of movable guards, high pressure etc. may have their own processing power and can be connected to the central control system by a communication bus.

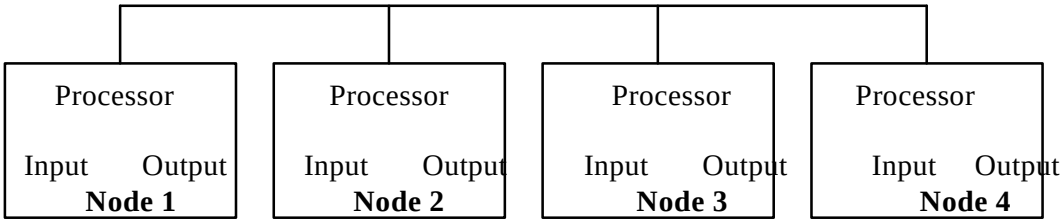


Figure 1 The principle of distributed control

The safety-related parts of the control system shall protect the operator from unacceptable risks. The EC machinery directive [MD] claims in annex 1, clause 1.2.1 ”Safety and reliability of control systems”

> *Control systems must be designed and constructed so that they are safe and reliable, in a way that will prevent a dangerous situation arising. Above all they must be designed and constructed in such a way that:*
>
> *- they can withstand the rigours of normal use and external factors,*
>
> *- errors in logic do not lead to dangerous situations.*

This is a very general requirement and it is not clear how to validate such a requirement, especially when a programmable electronic system is used to control the machine.

All safety functions of the control system must be covered by the safety validation. Errors in the logic of the safety functions must not be present. Correct behaviour is essential to guarantee the safety of the operator.

All safety-related parts of a control system should have their behaviour at fault specified. The behaviour can be divided into 5 categories according to figure 1. [EN954-1] Basic safety principles and well-tried safety principles should also be implemented for categories B, 1, 2, 3 and 4.

Category	**System behaviour**	**Principles to achieve safety**
B	The occurrence of a fault can lead to the loss of the safety function.	Mainly characterised by selection of components
1	As in category B, but the probability of occurrence is lower than in category B.	"
2	Faults are detected by periodic checks at suitable intervals. The occurrence of a fault can lead to the loss of the safety function between the checks. The loss of the safety function is detected by the check.	Mainly characterised by structure
3	When a single fault occurs the safety function is always performed. Some but not all faults will be detected. Accumulation of undetected faults can lead to the loss of the safety function.	"
4	When the fault (faults) occur the safety function is always performed. The fault (faults) will be detected in time to prevent the loss of the safety function.	"

Table 1.1 Categories for system behaviour at fault according to standard EN954-1

During 1996 SP did a safety study of distributed systems [SP9623]. The study aimed to document the faults and errors which are considered to be unique for such a system. The types of errors are as follows:

- **Node Errors**
  Transient or permanent hardware faults or software design faults may cause a node not to be fully operational. It is normally necessary to have a common understanding between all nodes in the network, which nodes are erroneous and which nodes are fully operational. This procedure is called membership agreement.

- **Bus Errors**
  The bus is vital in a distributed system. Two obvious types of errors are that messages are destroyed and that messages cannot be sent on the bus.
- **Timing Errors**
  The nodes may require fully synchronised and correct clocks in every node for correct operation of the system. Both hardware and software faults may result in incorrect timing.
- **Data Consistency Error**
  Nodescooperating on the same task should have data of the same age. Inconsistent data may lead to different decisions taken at the nodes, even if they are programmed with the same algorithms.
- **Initialisation and Restart Error**
  It will be hard to know in which order the computers of the network will start after a power up sequence. Proper routines for synchronisation must be implemented.
- **Babbling Idiot Errors**
  ”Babbling idiot” errors occur when one or several nodes in the system overloads the communication bus by erroneously sending a lot of high priority messages on the bus so that other nodes cannot send their messages.
- **Configuration Errors**
  Usually a system will only have the correct function if exactly the right types of nodes are used at the correct physical positions. An incorrect mix of modules, or an incorrect parametrisation of programmable modules, may cause a configuration error.

The use of distributed control gives a lot of advantages. Systems can be implemented by distributed nodes of standard type, a lot of cabling can be saved, the structure of the system is better etc. However, the safety must be able to prove (validate) also in a distributed system. Failures of safety-related functions in machinery may lead to injured operators or even death of operators. The safety level must be as high in a distributed system as in systems built using a central computer.

Much effort is spent within the standardisation organisations on how to validate the safety of machine control systems. Both the European CENELEC and the global IEC [IEC61508] are preparing standards which will be very important for the safety of machinery. This report has tried to adopt many of the basic ideas of the standard drafts, but it may have to be adjusted as the standards are issued.

2 Scope

Distributed control is used in several applications, e.g. process control, industrial automation, cars and mobile machines (e.g. excavators and dumpers). All applications are not safety-related. The validation method of this report focuses on safety-related applications for machinery, both permanently installed and mobile.

The validation methods presented are intended to be used for the validation of functional safety of distributed control systems.

Most validation methods do not sufficiently cover the faults that may be introduced by the use of communication buses. This new set of validation tools will supplement the ”established” methods. The conclusion on functional safety will be based both on the result of the general methods and on the result of the validation methods of this report. Both the general risks associated with programmable electronic systems, and the specific risks associated with the use of communication buses will then be covered.

The design and the safety validation must cover all aspects where safety can be influenced by the use of distributed control. This report concentrates on the validation method, even if several of the suggested techniques can be used already at the design phase.

3 References

[MD] ”The EC Machinery Directive”
Council Directive 89/392/EEC of 14 June 1989 on the approximation of the laws of the Member States relating to machinery as amended by Directives 91/368/EEC of 20 June 1992, 93/44/EEC of 14 June 1993 and 93/68/EEC (CE marking) of 22 July 1993.

[EN954-1] Standard EN 954-1:1996
Safety of machinery - Safety related parts of control systems
Part 1: General principles for design

[EN60204-1] Standard EN 60204-1:1998
Safety of machinery - Electrical equipment of machines

[IEC812] Standard IEC 812:1985
Analysis techniques for system reliability -
Procedure for failure mode and effects analysis (FMEA)

[IEC1025] Standard IEC 1025:1990
Fault Tree Analysis

[IEC61508] IEC 65A/264/FDIS
Draft IEC 61508-1 - Functional safety of electrical/electronic programmable electronic safety-related systems
Part 1: General requirements

IEC 65A/254/CDV
Draft IEC 61508-2 - Functional safety of electrical/electronic programmable electronic safety-related systems
Part 2: Requirements for electrical/electronic/programmable electronic safety-related systems

IEC 65A/269/FDIS
Draft IEC 61508-3 - Functional safety of electrical/electronic programmable electronic safety-related systems
Part 3: Software requirements

IEC 65A/265/FDIS
Draft IEC 61508-1 - Functional safety of electrical/electronic programmable electronic safety-related systems Part 4: Definitions and Abbreviations

IEC 65A/266/FDIS
Draft IEC 61508-1 - Functional safety of electrical/electronic programmable electronic safety-related systems
Part 5: Examples of methods for the determination of safety integrity level

IEC 65A/255/CDV
Draft IEC 61508-1 - Functional safety of electrical/electronic programmable electronic safety-related systems
Part 6: Guidelines on the application of Parts 2 and 3.

IEC 65A/256/CDV
Draft IEC 61508-1 - Functional safety of electrical/electronic programmable electronic safety-related systems
Part 7: Overview of techniques and measures

[IEC61496-1] Standard IEC 61496-1:1997
Safety of machinery -
Electro-sensitive protective equipment -
Part 1: General requirements and tests

[SP9623] Jan Jacobson, Lars-Åke Johanson, Magnus Lundin
Safety of Distributed Control Systems
SP Report 1996:23

[ISO 11898] Standard ISO 11898:19, Road-Vehicles - Interchange of digital information - Controller Area Network (CAN) for high-speed communication

[CANsyst] CAN System Engineering, From Theory to Practical Applications
Wolfhard Lawrenz, Springer-Verlag 1997
ISBN 0-387-94939-9

[CANKing] A CAN Kingdom, Lars-Berno Fredriksson, Rev. 3.01

[CANopen] CANopen Communication Profile for Industrial Systems, CAN in Automation, Rev 3.0

[CANalyzer] CANalyzer User's Guide, Vector Informatik GmbH, Ver 3.10

[IEC870-5-1] Standard IEC 870-5-1:1992, Part 5: Transmission protocols
Section 1 Transmission frame formats
Section 2 Link transmission procedures

[ISO7498] Standard ISO 7498:1984.
Information processing systems - Open Systems Interconnection - Basic Reference Model.

[Kop] A TTP Solution to an Automotive Control System Benchmark.
H. Kopetz, Institut fur Technische Informatik, Technische Universität Wien. 1994.

[TinBurns] Guaranteed Message Latencies For Distributed Safety-Critical Hard Real-Time Control Networks.
Tindell, K. Burns, A. Department of Computer Science, University of York. September 1994.

4 Definitions

Babbling Idiot
A babbling idiot is a node which overloads the communication bus, by erroneously sending a lot of high priority messages, such that other nodes can not send their messages. This fault will cause timing errors within the system.

CAN
CAN (Controller Area Network) is the most widely spread communication protocol for distributed control systems in mobile applications. Since CAN is a flexible and economic solution, it has also been used in many other applications.

Error
The manifestation of a fault in the system. Part of a system state which is liable to lead to a failure.

Failure
Deviation of the service delivered by a system from the specified service.

Fault
Error cause which is intended to be avoided or tolerated.

Fault Tree Analysis
To analyse what events, or combinations of events, that will lead to a hazard or serious consequence. (IEC 61508-7)

FMEA
To analyse a system design, by examining all possible sources of failure of a system's components and determining the effects of these failures on the behaviour and safety of the system. (IEC 61508-7)

Jitter
Jitter refers to time variations in actual start times of a process, as opposed to the stipulated release time. It is very important for sensor and actuator components that a maximum allowedjitter is guaranteed. In the periodic process model the allowedjitter can be indirectly specified by using the release time and the deadline. Jitter depends on clock accuracy, scheduling algorithms and computer architecture. Input and outputjitter can be used to relate thejitter of sampling and actuation processes respectively.

Membership Agreement
The procedure to get a common agreement within a group of nodes of a distributed system regarding which nodes that are operational. This group can be a minor number of nodes which are involved in a safety critical function, or it can be all nodes which are connected to the network. It is very important that the right actions are taken, in order to prevent from hazardous situations.

Safety
A measure of the probability that a system does not fail in such a way that that it either endanger human lives or puts high economical values at stake. This may be expressed as a probability of avoiding such a failure in an interval of a specified duration, given that the system was fault-free at the start of the interval.

Validation(for software)
The process of evaluating software to ensure compliance with specified requirements. (ISO 9000-3)

5 Validation principles

Functional safety of a machine can be described as the ability of the safety-related system to carry out the actions necessary to achieve a safe state for the machine, or to maintain a safe state for the machine. The functional safety shall be validated (proved) before a machine is taken into use.

Before a validation of any control system can be made, the machinery has to be identified and delimited. The functionality and the safety principles must be understood by the engineers who are to perform the assessment. The first step of the validation will then be to check if the required safety-related functions exist. All machines are affected by requirements from directives and standards. However, a design fault may adversely affect the required function. Even if the intention is to have the function correctly implemented, it will have to be validated. The validation will also cover the system behaviour at fault. Figure 5.1 illustrates the main questions.

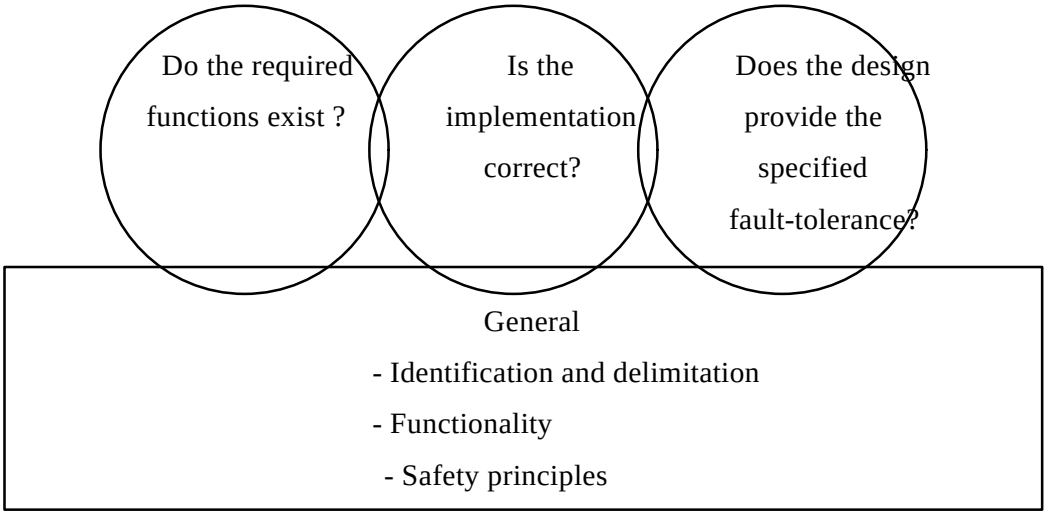


Figure 5.1 Questions during the validation

Validation is the process to prove that a product fulfils the requirements. If the system specification is good enough, start by validating the system specifications against the system requirements. The next step would be to verify the system implementation against the system specification by analysis and testing. The two steps may be performed as two separate actions of the validation work.

The validation may also be described as a sequential flow starting with the system aspects, continuing with detailed analysis of the design and ending with the test report. A validation at an independent test house may often be conducted in parallel to the development work. Such concurrent validation will reduce time-to-market for the developing company. An example of such a validation sequence is given in figure 5.2.

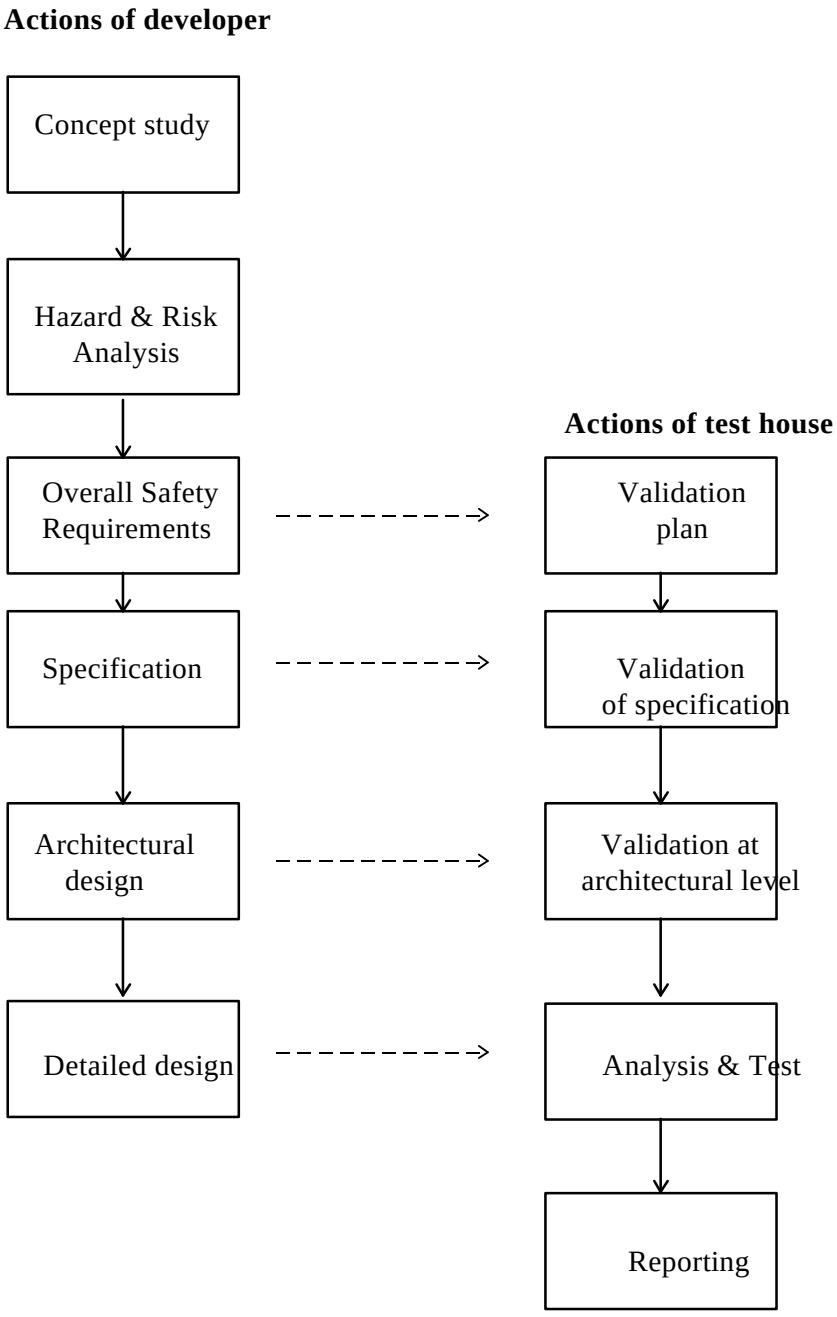


Figure 5.2 Example of a validation sequence

The objective of the validation cannot be to prove correct behaviour with 100% certainty. It is possible to achieve a high coverage, but not an absolute answer to the question of functional safety. A more realistic approach is to check the behaviour at fault, and that measures to achieve functional safety have been taken.

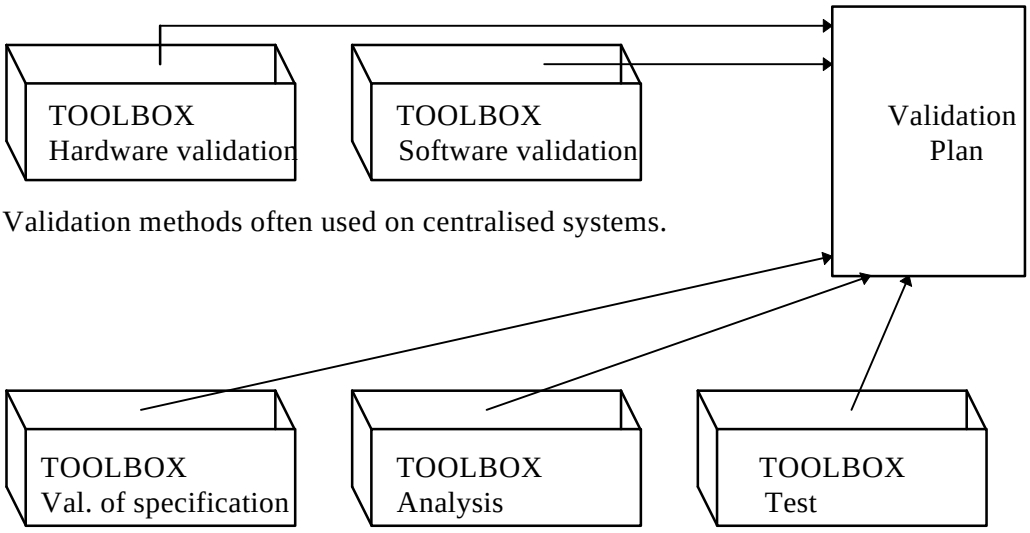
The developer will probably write a safety requirement specification which is to list both the safety functions and the safety integrity required for each function. [IEC61508]

Safety validation of programmable electronic systems has to cover both software and hardware aspects. Many validation methods exist which are employed for centralised systems, and may also be used for distributed systems. Examples of such methods are

software walk-through, hardware component FMEA and software call graphs. These methods have to be supplemented with methods specific for distributed control systems.

This report presents the validation methods for distributed systems in three chapters i.e. validation of specification, analysis and test. The intention is to create a toolbox from which validation methods can be selected. A complete validation program should consist of validation of specifications to prove that the intended functionality is correct, and analysis and testing to prove that the system behaves like it was intended to do.

Specific advice on which of the tools to select will not be given. This has to be judged in every separate case depending on the risks and the safety requirements. Examples of methods for validation of specifications concerning the bus and the communication between the nodes in the system are given. A number of analysis and test methods used to verify the behaviour of the system are also described.



Validation methods often used on centralised systems.

Validation methods for distributed systems.

Figure 5.3 Selection of validation methods

Other safety aspects include electrical safety and reaction to environmental stress such as electromagnetical interference (EMC). This report does not cover aspects other than those of functional safety.

Many of the validation methods may be slightly modified and used as design tools. The intention has been to concentrate on validation, and not to present a development methodology.

6 Validation of specification

The need for an accurate specification of a control system might not be obvious for all manufacturers. Many systems have traditionally been developed without exact, written specifications. Perhaps many of these systems have been so simple that the expected behaviour is obvious. The need for specifications increases as the complexity of a system increases and when (usually correlated) the possible degree of unexpected dangerous behaviour increases. A complex system is impossible to verify if there are no good specifications. It is impossible to guess the expected behaviour of different parts of the system. Complex systems have to be properly structured and specifications have to be present for all essential parts of the system.

A distributed system does not necessarily need to be more complex than a centralised system but the degree of possible malfunctions normally increases. This comes naturally from the increased flexibility that a distributed system normally gives the user. Furthermore the different parts of a distributed system might be manufactured by different companies. Therefore it is possibly true that distributed systems are more relying on good specifications than centralised systems do.

A good specification of a distributed control system covers all safety critical aspects of the system. This chapter gives information on how the specifications can be structured and what they shall contain in order to cover most aspects.

This issue of the report does not contain descriptions of all the validation methods.

Please contact SP, Ms. Lisbeth Pilgard (Phone +46 33 16 53 84,

Email lisbeth.pilgard@sp.se) to order the complete printed report.

7 Analysis

In many types of non safety critical electronic systems functional testing is the normal procedure to identify faults which have been designed into the system. A distributed system however is so complex that a functional ”black-box test” will only detect a very limited amount of the existing faults in the system. Therefore a theoretical analysis must be performed to be able to state if the distributed system is designed in a safe way or not. All design documents have to be accessible for the engineer who performs the test.

This chapter contains methods which can be used when analysing a distributed control system. It is best if the analysis can be performed concurrent with the construction of the system, since it can be very difficult to correct a fault which has been designed into the system. For instance, if a communication protocol without fault detection mechanisms is implemented, it can be very hard to rework the concept, in order to solve this weakness.

This issue of the report does not contain descriptions of all the validation methods.

Please contact SP, Ms. Lisbeth Pilgard (Phone +46 33 16 53 84,

Email lisbeth.pilgard@sp.se) to order the complete printed report.

8 Test

Tests of the distributed system can be made as a complement to the analysis. The following methods are examples of how a distributed system can be tested. The methods are generally known, and can be used in most distributed systems. Specific applications may require more tailor-made test methods in order to state if certain functions are reliable or not. To be able to do such a test, the test engineer needs deep knowledge of the application, and how the systems is intended to be used and possible misused.

This issue of the report does not contain descriptions of all the validation methods.

Please contact SP, Ms. Lisbeth Pilgard (Phone +46 33 16 53 84,

Email lisbeth.pilgard@sp.se) to order the complete printed report.

9 Conclusions

9.1 Development and validation

Distributed control systems should be designed in a way that enables safety validation. Different design principles are more or less suited to support the validation phase. There should be as little non-determinism and unspecified behaviour as possible, to make efficient validation possible. Everything that can be specified should be specified and documented to simplify the validation.

Many developers do not focus on the safety when developing distributed systems or modules. They are often satisfied when the system is working during normal conditions. Requirements for time-to-market and cost efficiency will often give priority to functionality. The safety aspects are not validated in most distributed systems.

9.2 Safety validation strategy

Safety must be designed into the product from the beginning. It cannot be handled as an add-on late in the development life cycle. The design for safety and the preparations for safety validation must be considered from the very beginning. This will be even more evident when working with high complexity systems, such as distributed control systems.

The aim of the validation work can not be to prove a design to be ”100% safe”. Much of the validation will be focused on the safety principles implemented to detect and handle faults in the system. The complexity of a distributed system will make it impossible to foresee all possible faults in the system. The ability of the system to handle faults will be very important.

Some machines are controlled by modules (nodes) developed by another company than the machine builder (the system integrator). The companies producing the modules will have to validate the proper behaviour, and the functional safety, of the modules. The system integrator will then have to handle modules from several suppliers to build an equipment of adequate safety level. There is a difference in the responsibility of the module developer and the system integrator. Detailed documentation of the design of all the modules will probably not be available at the overall safety validation. The module developer will always be responsible for his modules, and cannot deny responsibility by simply referring to compliance with a standard. The machine builder will always be responsible for the overall system safety.

9.3 Validating requirements from standards

The European standard EN 954-1 focuses on safety categories and system behaviour at fault. The categories state the required behaviour of safety-related parts of a control system in respect to its resistance to faults. Standard EN 954-1 also specifies that well-tried safety principles shall be used, without further specifying what this means for control systems. The validation methods of this report address both the behaviour at fault and the safety principles. Most of the methods are addressing the safety principles used in a distributed

system. An example of such a method is the analysis of membership agreement (see chapter 7). The method addresses certain safety aspects which, if they are implemented, give a considerable level of safety in case of faults within the membership group.

However some methods such as bus FMEA (see chapter 8), which can be used to validate if the system can withstand faults in the communication link, can be used to validate the behaviour at fault. Fault injections are made according to ISO 11898, while the system behaviour is studied.

Distributed control systems tend to become complex. In complex systems top down verification by functional testing often is very difficult. An identified wrong behaviour at the functional level might be very hard to couple to errors at lower levels. This is even harder if the functional errors are transient. Therefore, bottom up systematic analysis, testing and verification from the physical layer up to and including at least the communication handling layer, against well defined requirements often is to prefer. Therefore the safety assessment shall cover also the lowest level of a distributed system, like electrical driving capacity, bit timing and tolerances, filtering protection, etc. and the layers above covering the programming of the communication controllers and the communication handling software.

Since most of the methods address the safety principles of a distributed system, they can also be used as a guidance during development of the system instead of during validation afterwards.

9.4 Validation methods and fault types

It has been demonstrated that it is possible to select methods to check for faults in distributed control systems. As always in programmable electronic systems, it will not be possible to guarantee a fault-free design. The degree to which the validation is carried out will have to be specified for each system. There is no common agreement on how extensive the validation is required to be for a specific safety level. A minimum requirement for many systems will be to address all the unique faults types of a distributed control system by at least one validation method.

The different validation methods of this report are cross referenced to faults types in table 9.1.

Method	Node Error	Bus Error	Timing Error	Data Consistency Error	Init and Restart Error	Babbling Idiot Error	Configuration Error	Other
7.1.1				X				
7.1.2	X	X						
7.2.1	X	X						
7.2.2							X	
7.2.3							X	
7.3.1								X
7.3.2								X
7.4..1			X					
7.4.2			X					
7.4.3			X			X		
7.4.4			X					
7.5.1	X	X						
7.5.2					X			
7.5.3								X
8.1		X						
8.2	X	X						
8.3			X					

Table 9.1 Cross reference of validation methods and error types in distributed systems

9.5 Future development

The work in this project has shown that all organisations active in the field of distributed control systems have much in common. The equipment under control will differ, but the basic technique of distributed control is the same. Safety aspects are present in many of today's applications, and the number of safety-related applications for programmable control are bound to increase further.

There is a need to be able to validate functional safety in distributed control systems. Future research work should include more use of the validation methods on different applications.

Safety regulations and standards for safety of machinery and programmable electronic systems will develop further during the next few years. One example of this work is the IEC standard [IEC61508] on functional safety which is expected to be published in late 1998. The validation methods must interface to regulations and standards if they are to be useful for industry.