STUK-YTO-TR 127
JUNE 1996

# Programmable automation systems in PSA

**Urho Pulkkinen**
VTT Automation

In the Finnish Centre for Radiation and Nuclear Safety
the study was supervised by
**Harri Heimbürger**

This study was conducted by order of
the Finnish Centre for Radiation and Nuclear Safety

The conclusions presented in the report are those of the author and do not represent the official position of the Finnish Centre for Radiation and Nuclear Safety.

FINNISH CENTRE FOR RADIATION AND NUCLEAR SAFETY
P.O.BOX 14, FIN-00881 HELSINKI, FINLAND
Tel. +358-9-759881
Fax +358-9-75988382

ISBN 951-712-216-0
ISSN 0785-9325

Oy Edita Ab
Helsinki 1997

*PULKKINEN, Urho (Technical Research Centre of Finland). Programmable automation systems in PSA. STUK-YTO-TR 127. Helsinki 1997. 19 pp.*

**ISBN** 951-712-216-0
**ISSN** 0785-9325

**Keywords:** Safety, probabilistic safety analysis, reliability analysis, automation, programmable systems, reactor protection systems, nuclear reactor safety

# ABSTRACT

The Finnish safety authority (STUK) requires plant specific PSAs, and quantitative safety goals are set on different levels. The reliability analysis is more problematic when critical safety functions are realized by applying programmable automation systems. Conventional modeling techniques do not necessarily apply to the analysis of these systems, and the quantification seems to be impossible. However, it is important to analyze contribution of programmable automation systems to the plant safety and PSA is the only method with system analytical view over the safety.

This report discusses the applicability of PSA methodology (fault tree analyses, failure modes and effects analyses) in the analysis of programmable automation systems. The problem of how to decompose programmable automation systems for reliability modeling purposes is discussed. In addition to the qualitative analysis and structural reliability modeling issues, the possibility to evaluate failure probabilities of programmable automation systems is considered. One solution to the quantification issue is the use of expert judgements, and the principles to apply expert judgements is discussed in the paper. A framework to apply expert judgements is outlined. Further, the impacts of subjective estimates on the interpretation of PSA results are discussed.

*PULKKINEN, Urho (VTT Automaatio). Ohjelmoitavat automaatiojärjestelmät PSA'ssa. STUK-YTO-TR 127. Helsinki 1997. 19 s.*

**ISBN** 951-712-216-0
**ISSN** 0785-9325

**Avainsanat:** Turvallisuus, turvallisuusanalyysi, luotettavuusanalyysi, automaatio, ohjelmoitavat järjestelmät, reaktorin suojausjärjestelmät, reaktoriturvallisuus

# TIIVISTELMÄ

Suomen ydinturvallisuusviranomainen (STUK) edellyttää laitoskohtaisia todennäköisyysperustaisia turvallisuusanalyyseja (PSA). Lisäksi edellytetään eritasoisten kvantitatiivisten turvallisuustavoitteiden täyttämistä. Kvantitatiiviset luotettavuusanalyysi on vaikeaa silloin, kun jotkut keskeisistä turvallisuustoiminnoista on toteutettu ohjelmoitavalla tekniikalla. Perinteiset luotettavuusanalyysin menetelmät eivät tällöin sovellu sellaisenaan, ja kvantifiointia pidetään jopa mahdottomana. Kuitenkin on tärkeää arvioida ohjelmoitavan automaation merkitystä koko laitoksen turvallisuudelle, ja PSA on ainoa menetelmä joka kykenee tarkastelemaan analyyttisesti ja kokonaisvaltaisesti laitosten turvallisuutta.

Tässä raportissa tarkastellaan PSA menetelmien (vikapuuanalyysi, vika- ja vaikutusanalyysi) soveltuvuutta ohjelmoitavien järjestelmien tarkasteluun. Keskeinen ongelma mallintamisessa on, miten pieniin kokonaisuuksiin ohjelmoitava järjestelmä voidaan analyysissa dekomposoida. Kvalitatiivisten menetelmien lisäksi raportissa tarkastellaan mahdollisuuksia arvioida kvantitatiivisesti ohjelmoitavien järjestelmien vikaantumistodennäköisyyksiä. Ratkaisuksi esitetään asiantuntija-arvioita ja niiden yhdistämistä "kovaan evidenssiin" erityisen Bayesilaisen mallin avulla. Asiantuntija-arvioiden käyttö johtaa aina subjektiivisiin lopputuloksiin. Tämän vuoksi raportissa tarkastellaan myös subjektiivisuuden vaikutusta PSA:n tulosten tulkintaan ja käyttöön. Tämä ongelma ei liity ainoastaan ohjelmoitavaan automaatioon, vaan moniin muihin riskianalyysien erilliskysymyksiin (esim. inhimilliset virheet, ja tason 2 PSA:n epävarmuudet).

# CONTENTS

# 1 INTRODUCTION

Safety authorities require plant specific PSAs in many countries, and quantitative safety goals are set either on core melt frequency or safety function reliability level. The reliability modelling gets difficult when critical safety functions are realised by applying programmable automation systems. Conventional modelling techniques do not necessarily apply to the analysis of these systems, and the quantification seems to be impossible. However, it is important to analyse contribution of programmable automation systems to the plant safety and PSA is the only method with system analytical view over the safety.

This report concentrates on evaluating the applicability of PSA methods (fault tree analyses, failure modes and effects analyses) in the analysis of programmable automation systems. The difficulties of applying PSA methodology are connected with decomposing programmable automation systems for reliability analysis purposes, obtaining reliable quantitative failure data from testing or operational experience and using the partly qualitative evidence from various sources in estimating the system failure probabilities.

In this report the possibilities and limitations of usual PSA methods are discussed. As a solution to the quantification issue, the use of expert judgements is proposed and a framework to apply expert judgements is outlined. Further, the impacts of subjective estimates on the interpretation of PSA results are discussed.

# 2 PROBABILISTIC SAFETY ASSESSMENT (PSA)

The purpose of probabilistic safety assessment (PSA) is to give an overall view on plant safety by identifying the initiating events of accidents, describe the event sequences beginning from initiating events and leading to various plant damage states and radioactive releases. Further, PSA evaluates the plant risk quantitatively in terms of accident frequencies. PSA is usually divided in three levels. The level 1 PSA concentrates on the reliability modelling of emergency functions and accident sequence models, level 2 PSA considers the physical accident processes and the source term, and level 3 PSA evaluates the effects of radioactive releases on the environment. The focus of this paper is at level 1 PSA.

PSAs are used for several purposes. The can be applied to compare different designs from the safety point of view as well as to evaluate design modifications and backfittings. Further, they may be used to develop and evaluate procedures, such as safety related technical specifications and emergency procedures. PSA models can also be used as a common model between plant designers and users as well as between power companies and safety authorities.

On possibility to use PSA in licensing is to apply quantitative safety guidelines or safety goals, which approach has been adopted by some safety authorities. Safety goals are rules which specify acceptable risk for example by setting a limit for core damage frequency or safety function failure probability. If safety goals are set, then all types of hazardous events must be modelled and their probability must be estimated in order to check whether the requirements of the safety goal are met.

A recent way to apply PSA is so called living PSA, which means that the safety of the plant is monitored continuously by PSA. At the same time the PSA models are updated by using the information obtained from the plant.

PSAs are based on models. The reliability structure of the emergency functions are described by applying fault trees, which explain the failures of emergency functions as consequences of component failures, human errors and other basic events. The propagation of accidents beginning from initiating events and ending with various core damage states is described by event trees. The accident sequence frequencies are estimated by using statistical data on the basic event probabilities and initiating event frequencies. The fault tree and event tree models on one hand and statistical estimates for the basic event probabilities on another are the most important models of PSA level 1.

Conventional safety automation systems are usually included in PSA models. The degree of detail of the automation system models depends on the applications of PSA, and in this respect PSAs differ much from each other. Although the fault trees of automation systems may be rather extensive there are no philosophical difficulties in the modelling process: the automation systems do not differ from the other hardware systems.

The failures of programmable automation systems (as conventional automation systems or other hardware systems and human errors) are connected to accidents in several ways. They may cause initiating events and even so called common cause initiators, which cause both

initiating event and simultaneous unavailability of redundant safety functions. On another hand they can be system and component failures, which are intermediate or basic events of the fault tree models. These events cause the unavailability of safety functions or their components. The failures may be also latent, i.e. they have actually occurred in past before initiating event. As the failures of hardware components, also the failures of programmable systems may be dependent on each other. The common cause failures (CCFs) of programmable systems form a very difficult problem area.

The difficulties to include programmable automation into PSA-models are connected with the decomposition of the systems structure and with the problems of determining quantitative reliability estimates. The components or subfunctions of a programmable systems are not easily identified or described in such a way that they could be described easily as events of fault tree. Also the dynamic features of programmable functions cause problems. The quantitative reliability estimates needed in PSA, e.g. the probability of failure to operate properly when demanded, are not results of traditional software reliability models. In order to estimate these quantitative estimates statistically a large number of tests is needed.

In the following the possibilities to identify the failures of programmable systems by applying more or less conditional reliability engineering methods are discussed. Later, the reliability modelling and quantitative analysis are considered.

# 3 QUALITATIVE RELIABILITY ANALYSES AND RELIABILITY MODELS

Qualitative reliability analysis methods are applicable in identifying the failure modes of the system. They provide information useful for developing reliability modes and documenting the systems failure behaviour. One of the most wellknown reliability engineering method is the failure mode and effects analysis (FMEA), which is a standardised method (see IEC 812, 1985). It has been modified for the analysis of software by Reifer (1979) and developed further for example by Darricau and Hourtolle (1992). Software FMEA has been applied mainly in nonnuclear industry, e.g. in space applications. FMEA is used to identify the component failures, their causes and effects on the systems function. FMEA has been mainly applied to hardware systems, but it is applicable for the analysis of programmable systems. To perform FMEA, the systems function must be described first by identifying the functions of the component. The objective of the analysis is to identify potential failures in the components, and the consequences these may have on the system. In this way it is possible to identify the potential failures which are most safety critical, and therefore require the most attention. FMEA is a bottom up analysis, starting with the components, and gradually increasing the scope to analyse total functions.

The application of FMEA for analysis of programmable systems requires some modifications. The modified FMEA form includes in addition to the usual information a more exact description of the failure mode, which may be divided into hardware and software failure modes. The hardware failures are described separately from software failure modes in order to emphasise the different nature of the software faults. The dynamic software functions require more attention, and it is not clear whether FMEA can really identify all failures.

It is not practical to apply FMEA to the analysis of software structures. The analysis is most effective on the level of functions of the different modules of software based systems. These modules may not be identified without preliminary fault tree analysis, which suggest that it is advantageous to apply it iteratively with reliability modelling. This kind of iterative approach decrease also the resources needed in the analysis.

Sneak Circuit Analysis, SCA (Taylor, 1992), is a combination of FMEA-type approach and fault tree. In SCA the output of the software, which may lead to hazardous events are identified. Further the causes of errors are studied. The findings of the identification are described by fault trees.

Methods like Preliminary Hazard Analysis, PHA, Software Requirements Hazard Analysis, SRHA, Subsystem Hazard Analysis, SSHA, System Hazard Analysis, SHA, and Operating & Support Hazard Analysis, O&SHA, are based on the conventional hazard analysis methods (see McKinlay, 1991). The method developed by Toola (1992) for the analysis of accidents and disturbances of process automation belongs to the same family of methods. The above methods have been applied in nonnuclear industry, where extensive risk analyses are not usually made. In fact, it is likely that the analyses made in connection to plant specific PSA provide the same information as the above mentioned methods. However, these approaches

reveal weaknesses in the interface between the software and the other parts of the system (see Pulkkinen, 1993).

In PSA environment, FMEA and other qualitative analyses are applied in order to form a basis for fault tree models. Further, it is a documentation of identified failure modes which are included into fault trees as basic events or fault tree gates. In the case of conventional systems, it is rather easy to identify the failure modes at each level of functionality (i.e. at system, subsystem, and component levels). This means that the conventional hardware systems may be decomposed into components or subsystems with respect to the failure effects. This is more difficult in the case of programmable systems. From the hardware point of view, it is rather easy to describe the failures of each electrical unit and to identify the consequences of failures. However, the application of software based systems makes it possible to integrate many functions into one electrical unit (e.g. CPU), and the functionality of software doesn't correspond directly to the hardware units.

When a programmable system is decomposed for reliability modeling purposes, it should be done with respect to the systems emergency functions. This means that such failure modes, which cause the loss of the emergency functions are included into the model. Problems may arise when the system consists of redundant channels. Usually redundant channels can be seen as independent and they are modelled as inputs to an "AND"-gate in the fault tree. This requires that one must be able to judge how independent the channels really are. If the channels are dependent, they may utilise common processors or common service functions. In principle this kind on dependencies can be described by fault tree models: they correspond to the shared equipment (or shared function) dependencies usually encountered in PSA-modelling. The other dependencies, i.e. common cause failures caused by simultaneous failures of either software located in redundant channels or hardware can only be modelled as parametric common cause failures, i.e. they are taken into account only as parameters, which increase the probability of multiple failures.

There are many functionalities, which are specific to programmable systems. These systems may utilise extensively selfdiagnostic functions and "software based redundancy". The latter means that some functions (within a redundant channel) are made more reliable by implementing them with several redundant software units, which perform more or less independently the same task. Both of the above functionalities increase the reliability, but it is not clear how this can be modelled in fault trees. However, since this redundancy is implemented in a special or even dynamic way, it seems feasible not to describe it in fault trees. From the practical point of view, the modelling of this redundancy would make the fault trees too complicated. From more theoretical point of view, it is not possible to model dynamic features of software by static models, such as fault trees. The same principles apply partly also for the description of the diagnostic functions.

The software of programmable systems may be divided to system software and application software. The system software includes for example the operating system, and the application software is designed to produce the functions of the system. In PSA, the most important issue is to model the loss of emergency functions and its dependency on component failures. At the level, which is important for PSA, high degree of detail of the programmable systems is not needed. The division to system and application software is not always needed from that point of view. As a conclusion, it is recommendable to model only the most important functions of the programmable systems.

# 4 QUANTITATIVE RELIABILITY ANALYSES

If the reliability structure of a programmable system is established and described by a fault tree model, it is possible to determine the minimal cut sets and thus find the most vulnerable points of the system. The information contained by the minimal cutset list is most useful: it gives a qualitative view over system in a systematic way and it gives guidelines to direct the more extensive analyses to most critical issues.

However, in many cases it is advantageous to evaluate quantitatively the systems reliability. This is necessary, when the safety authority has set quantitative safety goals. Further, it is often important to know what is the importance of certain programmable systems for the accident frequency. It is worth noticing that this may imply also that one must determine the failure rate of such programmable automation systems, which are not emergency systems but the failures of which may cause initiating events.

The reliability models of PSA require various types of statistical reliability data. Depending on the system and its components estimates for operating time failure rates, standby time failure rates, repair times, component unavailabilities, probabilities to fail at demand or parameters of common cause failure models are needed. Often the evidence from operational experience is not sufficient for the statistical estimation of the above mentioned parameters, even in the case of commonly used components. This is a problem especially when the PSA is required to correspond the plant specific operational experience. To solve the problems due to lack of reliability data, many approaches are applied. One possibility is to used generic data bases, which may not actually correspond to the plant in question. Another way is to use expert judgements, for which purpose there are methods. Independently from the source of reliability data, PSA-methodology makes it possible to analyse the uncertainty of the risk estimates cause by the uncertainty of the reliability data and other parameters. The possibility of consistent uncertainty analyses makes PSA transparent also in this respect.

The reliability data for programmable components seems at the first sight rather similar to that of conventional systems: the problem seems to be only on the sufficiency of statistical evidence. However, this is not the whole truth. Some of the above mentioned reliability parameters may not be relevant for programmable systems. For example, the probability of failure when demanded is a common parameter for the reliability of emergency system. In the case of conventional system, the demand is exactly defined for example as manual start signal. The programmable systems operate cyclically, and they measure their inputs at certain time points. If there is a demand, i.e. the input corresponding to manual trip signal is true, the system should operate. However, at each time point where the system reads its inputs there is also a demand, which requires certain function. One may ask is the definition of the event "failure at demand" really similar for programmable system and conventional system. Same kind of problems may be actual also for other parameters.

Another issue, which must be noticed is that the reliability of programmable systems is a property of the systems operation environment as well as that of the system itself. Although there are errors in the software, these errors may cause a loss of safety function only in the case

of inputs which occur with very low probability during demands. In other words, the reliability of programmable systems depend on the operational profile, which as the probability distribution of input sequences, varies from one environment to another. This restricts the use of generic operational experience in determination of reliability parameters.

Quantitative reliability estimates are always based on certain evidence, which is most often operational experience statistics. For programmable systems this evidence is either very limited or not applicable due to differences between the operational profiles of the data source and the actual system. Another source of evidence is obtained from the dynamic testing of the system. If high reliability is required, the number of tests is very large, and the it may be practically impossible to test the system so extensively. Thus evidence from other sources must be used in order to estimate the reliability parameters.

The statistical information gathered during the development process of programmable systems is related with the systems reliability, and in principle it may be applied in reliability estimation. In order to use this information, software reliability growth models can be applied. However, it is not directly possible to evaluate the weight of these measurements with respect to the PSA estimates.

The abovementioned development process follows certain quality assurance and quality control principles, which are based on applicable standards, but which may vary from one developer to another. More strict principles are believed to result to more reliable products. Thus the quality assurance process provides evidence on reliability. Same may apply to other tools and principles followed during the development process. To use this kind of evidence in determining the reliability estimates requires models and ability to weight the evidence.

The use of several kinds of evidence in determining reliability estimates is possible only through expert judgements, since validated models and extensive empirical observations are not available. As already mentioned, experts judgements are almost routinely applied in PSA to determine for example human error probabilities or describing uncertainty on some critical physical phenomena. The methodology for use of expert judgements is, however, under development and research on this topic is still going on. A framework for applying expert judgements in quantitative reliability analysis of programmable systems is outlined in the following section (see e.g. Pulkkinen, 1994, Cooke, 1991).

Since expert judgements are based partly on subjective assessments and weightings, it is not possible to interpret the results of PSA as objective. Thus the subjective interpretation of probability must be accepted. The implications of using subjective probabilities are discussed also in the following section.

# 5 USE OF EXPERT JUDGEMENTS IN QUANTITATIVE RELIABILITY ANALYSIS

Expert judgements are applied to analysis of many issues of PSA. Typical examples are evaluation of human error probabilities and analysis of complicated physical phenomena. Common to the issues analysed on the basis of expert judgements is that there is the lack of experimental data and the validated models. Further, the experts on the issue do not fully agree the nature of the issue.

The proces of applying expert judgements involves the selection of issues for the analysis, selection and training of experts, elicitation of expert opinions, combination of estimates and reporting of the analysis. Depending on the method and case under analysis, the extend of these phases may vary. The literature of using and modeling expert judgements is wide (see e.g. Reiman, 1994, Pulkkinen, 1994, Pulkkinen & Holmberg, 1997, Mosleh & Apostolakis, 1984). One of the most important applications of expert judgement methodology can be found from the NUREG-1150 (1989) study.

## 5.1 A Bayesian framework for using expert judgements

To obtain better estimates for reliability of programmable systems, all possible evidence should be applied in the analysis. As discussed earlier, this require extensive applications of experts opinions about the weight of various pieces of evidence. A most suitable approach for using analysing experts judgements is based on Bayesian models. To outline and illustrate such model, the failure probability $(\theta)$ of a programmable system is considered. It is assumed that $\theta$ depends on certain factors, which are denoted by $\boldsymbol{X} = (X_1,...,X_n)$, in a stochastic way. The exact dependence on these factors is unknown, and it is modelled by a conditional distribution of $\theta$ given $\boldsymbol{X}$, or $p(\theta \mid \boldsymbol{X})$. Another possibility to describe the relation between $\theta$ and $\boldsymbol{X}$ is to model $\theta$ and a function of $\boldsymbol{X}$ and random disturbance or noise. This kind of model leads, however to the conditional distribution $p(\theta \mid \boldsymbol{X})$. The stochastic nature of the relation between $\theta$ and $\boldsymbol{X}$ corresponds to the fact that the same development procedures do not lead automatically to same reliability, or equivalently, the attributes $\boldsymbol{X}$ don't determine the reliability uniquely and deterministically. The factors $(X_1,...,X_n)$ describe for example the development process, quality control principles, etc. Part of the factors may be directly observable and measurable, part may be unobservable or qualitative characterisation of the system and its development process. The framework is described in more detail by Pulkkinen (1994), Pulkkinen & Pyy (1996), and Pulkkinen & Holmberg (1997).

## 5.2 A specification of an expert judgement model

The model relating $\theta$ and $\boldsymbol{X}$ is usually parameterized, i.e., the distribution $p(\theta \mid \boldsymbol{X})$ depends on some parameters

$$p(\theta \mid X) = p(\theta \mid X, \alpha), \tag{1}$$

in which $\alpha$ is a parameter vector. The form of the model (1) and the interpretation of the parameters should be such that experts can give their assessments with respect to the model. Simple and transparent model is for example the logistic regression

$$\ln\left(\frac{\theta}{1-\theta}\right) = \alpha_0 + \sum_{i=1}^{n} \alpha_i x_i + \omega, \tag{2}$$

in which $\omega$ is a random (unknown) noise. It is possible to deduce the conditional distribution (1) from (2), when the statistical properties of the noise term are modelled by suitable distribution.

The experts are asked to evaluate the parameters ($\alpha$), and if necessary, the factors $(X_1,...,X_n)$. These assessments form the evidence obtained from expert judgements. The assessments concerning $X$ are denoted by $Z_i$, $i=1,...,m$ and the assessments concerning $\alpha$ by $Y_i$, $i=1,...,m$, in which $m$ refers to number of experts. In addition to the experts evidence, there may be results from the testing of the system, for example in the for "$k$ failures occurred in $r$ tests".

These evidence form tests and expert judgements can be combined by applying a Bayesian model. To do this, one has to specify the conditional probability of having $k$ failures in $r$ test given $\theta$ and the conditional (joint) distributions of $Z_i$ and $Y_i$ given $X$ and $\alpha$. The interpretation of these distributions is important. They are models of experts' ability to produce estimates for unknown variables. They should be as simple as possible to keep the model transparent. Models like the additive or multiplicative error models discussed by Mosleh and Apostolakis (1988) are suitable for this purpose. The whole expert judgement model can be expressed as Bayesian network. For the above simple model the Bayesian network has the form presented in Figure 1.

Figure 1 specifies the joint distribution of all variables of the model, and describes the dependencies between the variables. When the test results and experts' judgements become available, it is possible to determine the posterior distribution of $\theta$. The posterior distribution, $p(\theta|(k,r), Z_i, Y_i, i=1,...,m)$ combines the whole evidence on $\theta$ together. The numerical procedures for calculation of posterior distribution can be solved for example by applying Monte-Carlo methods (see Pulkkinen, 1994).

Generally, it is possible to combine various assessments of the system reliability to obtain better reliability estimates. The model structure depends on the reliability parameter and the form of evidence. The Bayesian approach provide a transparent way to model and determine the distributions of unknown variables. The modelling task is without doubt very difficult, and one must be aware of the nature of interpretation of the model predictions.

When the reliability parameters for all events corresponding to programmable systems are determined, they can be used as input data for

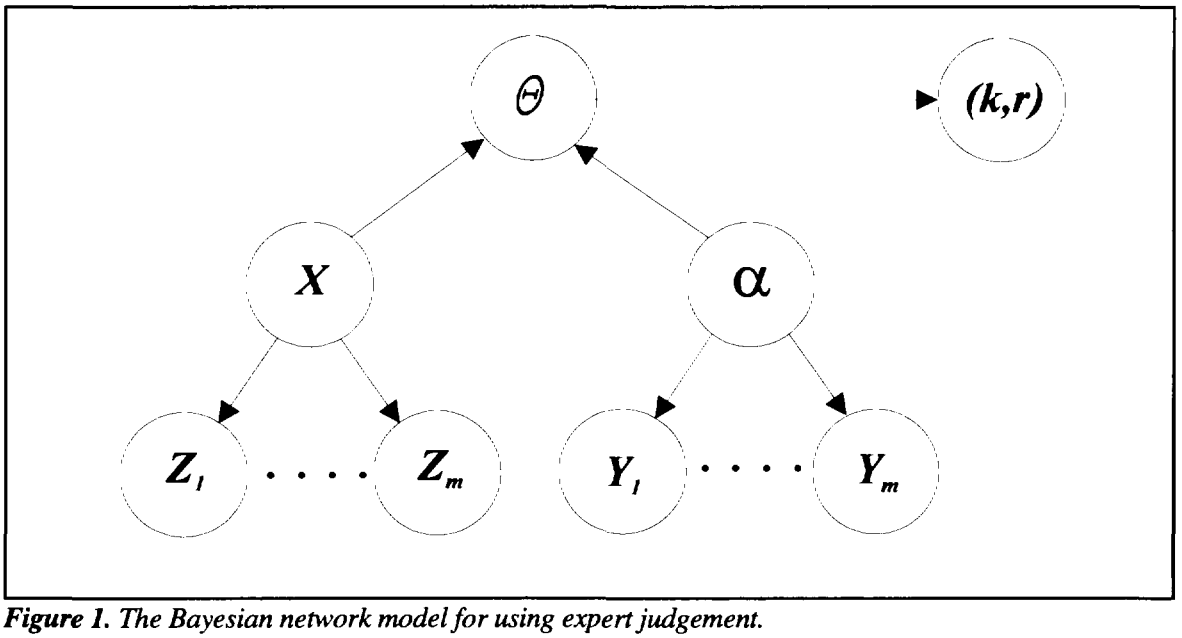


***Figure 1.*** *The Bayesian network model for using expert judgement.*

fault tree calculations. Since the Bayesian approach automatically leads to distributions of parameters instead of point estimates, the uncertainty of the estimates is expressed consistently. The posterior distributions can also be applied in the uncertainty analyses of PSA-models.

The above model is general, and to make it practical we must consider the factors $X$ in more detail. First, there must be a possibility to measure the factors, at least using a nominal scale. Secondly, there must be information on these factors, at least in the form of experts' judgements. Thirdly, the factors must have a clear interpretation, which the experts can accept and understand.

In principle, the experts should select the factors by themselves. However, we believe the factors are connected with 1) the software development process, 2) the management of the development process, 3) the quality assurance of the software, 4) documentation of the software. In addition of these factors, the operating experience should be taken into account.

The software development process is based on the process description, which defines e.g. the phase of the process and the results of each phase. It is advantageous that various standards are applied when the process is defined. In the expert judgement model, one could define variables measuring (at least on nominal scale) the degree of normalisation of the process and the use of standards.

The management of development process comprises the process design and follow-up. It requires that the project organisation is explicitly described and the tasks of various parties are defined. Further, it requires that the process is controlled and measured with certain indicators. In the expert judgement model, one must define variables describing the formality of the management process.

The quality assurance is a concentrated on the final result of the development process (i.e. the software based system). The quality of the end result is controlled and monitored with various numerical indicators, some of which may be used directly in the expert judgement model. Examples of such indicators are the number of audits, number of deficiencies or faults found in audits and failures found in testing.

The reliability of a software product depends on the quality of documentation of the development process phases. A good documentation system includes standardized report templates, and clear tools for product description and it provides a variety of checklists for several purposes. The documentation itself consists of product descriptions. In the expert judgement model, one should define variables measuring the degree of formality of the documents and the quality of the documentation system.

It is advantageous to organise tha above factors into a hierarchy, which helps their measurement and asseesment of their relative weights. Further, hierechic structure makes it possible to combine the evidence from expert judgements with hard evidence from tests and operational experience.

# 6 INTERPRETATION OF PSA RESULTS

PSA is always based on assumptions and judgements, which are in most cases partly subjective. This implies that the results of such PSA should be used with care and certain principles should be followed. This is the case especially when the reliability of programmable is analysed quantitatively and expert judgements are used as evidence on the reliability. The reliability estimates based on expert judgements are always subjective, which may restrict the use of PSA in licensing and decision making. As subjective assessments of plant safety, the results of PSA express the uncertainty of the PSA team about the accidents and their consequences. The probabilities and accident frequencies are rather the analysts degree of beliefs on the plant than real properties of the plant itself.

When reliability estimates based on expert judgements are applied in licencing, it is often required that the estimates should be conservative. This may be interpreted in two ways. First, it is possible to think that a failure probability estimate is conservative, if it is some kind of an upper bound. The other possibility is to say a failure probability estimate conservative when it is uncertain or based on weak evidence. The first interpretation may overemphasize the safety signifigance of the software based system. The second alternative requires that the uncertainty of the estimates is evaluated and thus the weight of evidence is also considered.

PSA results may also be used in other purposes, e.g in prioritizing further analyses or backfitting. In this kind of applications one should use as realistic estimates as possible. In this respect the interpretation of conservativity as large uncertainty is a more proper approach (from a Bayesian point of view).

If the assumptions and models are realistic, not conservative, and the uncertainty is expressed in a proper way, then the results provide guidance for decision making. They help in identification of the most important accident contributors, and both the safety authority and the licensee can base their decisions on these results.

The PSA results reflect the modeling assumptions and the strength of evidence behind the risk estimates. Given that the assumptions and evidence is in some sense essentially subjective, it is not recommendable to base the licensing decisions totally on the quantitative PSA results. A more proper and practical approach is to consider whether the modeling assumptions are realistic and whether the different sources of evidence are weighted in acceptable way, which implies also that one has evaluated whether the uncertainty about different phenomena is taken into account.

The quantitative safety goals have specific role when subjective (Bayesian) interpretation of PSA is followed. When the analysts try to meet the safety requirement, they at the same time try to find such assumptions or such pieces of evidence which lead to results compatible with safety goals. The next step is to analyze if these assumptions are credible or realistic in the case of the plant under analysis.

# 7 CONCLUSIONS

This paper considers the possibilities to include programmable systems into PSA models. Although the programmable systems are in many respects fundamentally different from conventional systems, it seems feasible to analyze them by usual PSA methods. Qualitative methods, suchs as FMEA, are useful in identifying the failure modes of programmable system. However, it is not advantageous to go into too deep detail. It is not probable that the dynamic aspects of software based systems are tackled by such, basically static analysis methods. Similarly, it is possible to create fault trees describing programmable systems, but is not possible to model them in deep detail. A suitable decomposition seems to be at electrical unit level. this means that, for example, redundant channels of emergency signals may be modeled.

The qualitative results of reliability models and qualitative analysis are included in the minimal cutset list, which actually identifies the weak point of the system with respect to system failure of accident sequence. This list helps in identifying the features of the systems, for which deeper analyses are needed. These deeper analyses should be made by applying methods, which are more suitable for programmable systems (i.e. formal methods, etc.).

The quantitative analysis requires reliability data for components, which usually do not exist in experimental of statistical form for programmable systems. In estimation of reliability, evidence from systems test and expert judgements concerning the system and its design must be used. To combine this evidence, the application of Bayesian methods seems to be a feasible approach. However, it leads to subjective interpretation or PSA results, which has implications on the use of PSA in decision making and licensing. Keeping in mind the limitations of PSA methodology, the interpretation of more or less subjective results and the special features of programmable system, it is useful to apply PSA although part of the plants emergency systems are based on programmable technology.

# REFERENCES

Cooke RM. Experts in Uncertainty. Opinion and Subjective Probability in Science. *Oxford university Press,* New York 1991: 1–321.

Darricau V, Hourtolle C. (1992) Application de l'analyse des effects des erreurs du logiciel (AEEL) a un logiciel embarque spatial: methode, resultats, enseignements. *In Petersen KE, Rasmussen B. (eds.) Safety and Reliability '92 Proceedings of the european Safety and Reliability Conference '92,* The first Pand European conference on safety and reliability under auspices of ESRA (European Safety and Reliability Association, CEC), Copenhagen, Denmark 10–12 June 1992: 739–750.

IEC 812. Analysis techniques for system reliability—Procedure for failure mode and effects analysis (FMEA). 1985: 1–41.

McKinlay A The State of Software Safety Engineering. In *Apostolakis G (ed.) Probabilistic Safety Assessment and Management, Elsevier,* New York 1991: 369–376.

Mosleh A, Apostolakis G. Models for the Use of Expert Opinions. *In Waller RA, Covello VT (eds.) Low-Probability, High-Consequence Risk Analysis, Plenum Press,* NY 1984: 107–124.

NUREG-1150. (1989) Reactor risk reference document. Washington D.C., U.S. Nuclear Regulatory Commission, Office of Nuclear Regulatory Research, 1989. (Report NUREG-1150).

Pulkkinen U. (1993) Reliability analysis of software based safety functions. *STUK-YTO-TR 53.* Helsinki 1993: 1–64. (In Finnish).

Pulkkinen U. Statistical Models for Expert Judgement and Wear Prediction. *VTT Publications 181. Technical Research Centre of Finland,* Espoo 1994: 1–65 + app. 80 pp.

Pulkkinen U, Holmberg J. A method for using expert judgement in PSA. *STUK-YTO-TR 129.* Helsinki 1997: 1–32.

Reifer DJ. Software Failure Modes and Effects Analysis. *IEEE Trans. Reliability, vol. 28 no. 3.* 1979: 247–249.

Reiman L. Expert judgement in analysis of human an organisational behaviour at nuclear power plants. *STUK-A118, Finnish Centre for Radiation and Nuclear Safety,* Helsinki 1994: 1–226.

Taylor JR. (1992) Human Reliability and Software Safety Analysis for Command and Control Systems. *In Petersen KE, Rasmussen B. (eds.) Safety and Reliability '92, Proceedings of the European Safety and Reliability Conference '92,* The first Pan European conference on safety and reliability under auspices of ESRA (European Safety and Reliability Association, CEC), Copenhagen, Denmark 10–12 June 1992: 762–771.

Toola A. Safety analysis in conceptual design of process control. *Technical Research Centre of Finland, VTT* Publications 117, Espoo 1992: 1–100.