Safety case framework to provide justifiable reliability numbers for software systems

Jan-Erik Holmberg[a*], Peter Bishop[b], Sofia Guerra[b], Nguyen Thuy[c]
[a]VTT, Espoo, Finland
[b]Adelard, London, UK
[c]EDF R&D, Chatou, France

Abstract: Very high reliability figures cannot be formally justified for a piece of software. Failure probabilities lower than 1E-4 are rarely claimed or justified even in a highly diversified software system, and there is not an accepted approach for the use of quantitative evaluation for software reliability between the different countries. The situation is even more difficult concerning figures for software CCF. The current state of the art for the quantification of software reliability relies mostly on holistic approaches, such as conformance to appropriate safety standards such as IEC 61508, or statistical testing. The EU Euratom FP7 project HARMONICS (Harmonised Assessment of Reliability of Modern Nuclear I&C Software) will tackle the problem of software reliability quantification using analytical and Bayesian approaches that take into consideration all the information available, in particular evidence obtained by V&V. Key to these approaches is how different pieces of evidence are interpreted in a probability model context and how their interrelationships are assessed. This can be combined with other approaches that model the development process and use development fault data to estimate the number of residual faults. This information can then be used to estimate worst-case bounds on the software reliability. The justification of the reliability estimated will follow the concept of a structured safety case, which is a solution to get a coherent process for the quantitative reliability assessment of software-based systems.

Keywords: Software reliability, safety case, Bayesian belief network

1. INTRODUCTION

The reliability and safety of computer-based systems that implement safety functions are critical issues for the construction and modernisation of nuclear power plants. This is in particular due to the fact that software can usually not be proven to be defect-free, and that postulated residual defects could be suspected of leading to common cause failure that could defeat redundancy and defence-in-depth. Unfortunately, the differences in current safety justification principles and methods between different countries restrict co-operation and hinder the emergence of widely accepted best practices. They also prevent cost sharing and reduction, and unnecessarily increase licensing uncertainties, thus creating a very difficult operating environment for utilities, vendors and regulatory bodies.

Given the experience with nuclear-related and software-based systems worldwide, there is now the possibility of using empirical reliability data in a way that has not been feasible before. In addition, advances in computer power and testing techniques means that simulated experience and statistical testing are becoming more practicable as forms of evidence. This evidence could have an important role in the assurance of nuclear I&C systems. Advances have also been made, and practical experience gained, in several other domains, such as the formal verification of software, defensive measures to tolerate postulated residual software faults, and safety justification frameworks.

This paper describes general approaches to be investigated by the HARMONICS (Harmonised Assessment of Reliability of Modern Nuclear I&C Software, 2011–2014) project to provide justifiable reliability numbers for software of computer-based safety systems in nuclear power plants. These numbers may be used to represent the software-related aspects of the computer-based systems in probabilistic safety assessments (PSA). Since the project is still in the initial phase, only the outline and principles of the approaches, which are based on the safety case framework and may include Bayesian belief network (BBN) modelling, are presented here.

2. HARMONICS OVERVIEW

The overall objective of the HARMONICS project is to ensure that the nuclear industry has well founded and up-to-date methods and data for assessing software of computer-based safety systems. It will take advantage of the aforementioned advances to propose systematic and consistent, yet realistic and practical approaches for software verification, software safety justification and quantification of software failure rates.

The project will address three key issues: software verification & validation (V&V), software safety justification, and quantitative evaluation of software reliability (i.e., software-related aspects of system reliability). The focus will be mainly on the independent confidence building for software of I&C systems implementing Category A functions (as defined by IEC 61226 [1]), which is the highest safety category in nuclear power plants (NPP), see Figure 1.

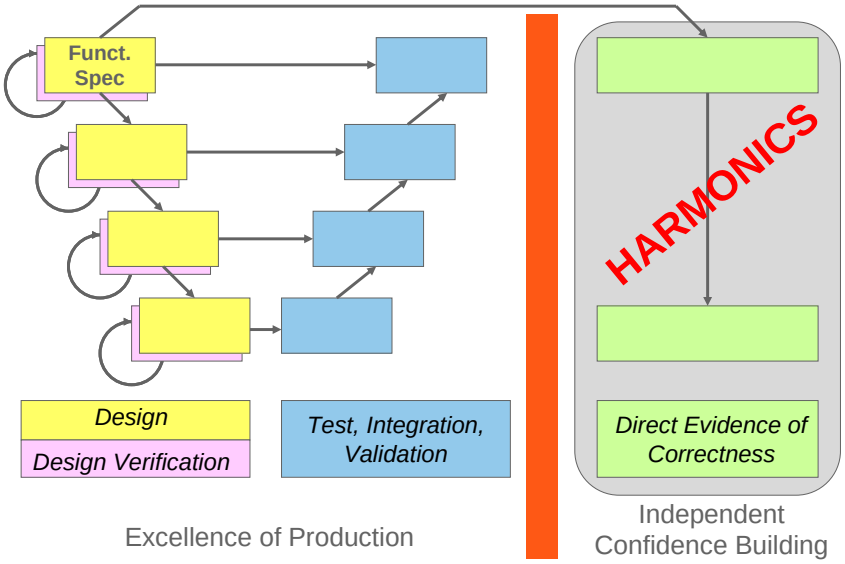

Figure 1. HARMONICS perspective on verification

Regarding software safety justification, the HARMONICS project will build on current practices and on results of previous Euratom FP6 research projects, namely CEMSIS (Cost-Effective Modernisation of Systems Important to Safety) [2] and BE-SECBS (Benchmark Exercise on Safety Evaluation of Computer Based Systems) [3]. In particular, it will propose a framework integrated into the overall system safety justification, and based on the complementarity and integration of the rule-based, the goal-based and the risk-informed approaches. The project will analyse the domain of applicability and acceptability of each approach, and will provide practical guidelines based in particular on the information gathered with the proposed V&V techniques.

HARMONICS intends to integrate quantitative software reliability claims in the overall software and system safety justification. In particular, the project will investigate the nature and justification for any reliability claim limit. Very high reliability figures cannot be formally justified for a piece of software, partly due to the pragmatics of the evidence required to justify such figures (e.g., the number of test required to justify the reliability), partly due to the epistemic concerns (e.g. how representative tests are of the operating profile). Failure probabilities lower than 10^{-4} are rarely claimed or justified even in a highly diversified software system [4], and there is no consensus on failure data for modelling of software failures [5]. The situation is even more difficult concerning figures for software common cause failures (CCF) modelled e.g. by beta-factors in PSA. The current state of the art for the quantification of software reliability relies mostly on holistic approaches, such as conformance to appropriate safety standards such as IEC 61508 [6], or statistical testing.

HARMONICS will tackle the problem of software reliability assessment using analytical approaches that, take into consideration all the information obtained by V&V. One of the possible ways of organising the various pieces of evidence in a probabilistic format is to use Bayes belief network (BBN). BBN has been advocated in this context also in [3], [7], [8], [9].

3. SAFETY CASE FRAMEWORK

3.1. Overview

A Safety Case is a structured argument, supported by a body of evidence, that provides a compelling, comprehensible and valid case that a system is safe for a given application in a given environment [10]. Current safety case practice uses a basic approach where claims are supported by evidence and a "warrant" that links the evidence to the claim.

The actual claim decomposition and structuring is normally very informal and argumentation is seldom explicit. In practice, the emphasis is on communication and knowledge management of the case, with little guidance on what claim or claim decomposition should be performed. The "case" and associated supporting tools can be seen as having two main roles:

- Reasoning and argumentation: as an over-arching argumentation framework that allows us to reason as formally as necessary about all the claims being made.
- Negotiation, communication, trust: as a boundary objective between the different stakeholders who have to agree (or not) on the claims being made about the system.

One approach is to explain safety assurance in terms of a "triangle" comprising rule-based, goal-based and risk-informed approaches. In the goal-based approach, the applicant develops a safety case in which the claims, evidence, and arguments are presented explicitly. This approach aims at providing a practical and convincing justification that certain safety properties are satisfied. The rule-based approach justifies compliance of software with a set of laws, rules and standards. A risk-informed approach represents a viewpoint to establish claims and requirements that are focused on safety justification on design and operational issues. This approach can be used to provide the basis for additional requirements or claims.

None of these approaches is without problems. The goal-based approach requires ensuring that the initial set of goals is complete and coherent. The rule-based approach is not enough on its own since it does not by itself often demonstrate credibly that a system is safe enough for a given application. Every approach includes features of objectivity and subjectivity, but the most subjective approach is the risk-informed approach based mainly on practiser's talent and expertise, which add uncertainty of the results (e.g. uncertainty of failure modes).

The aim of HARMONICS is to improve safety justifications by integrating these three approaches (goal, rule, and risk-informed) to get a coherent process for justifying software-based systems. The exact form of the safety case based on these approaches depends on the application and on negotiation by the parties involved, but the application specific subjects must conform to general form of safety case. The project will analyse the domain of applicability and acceptability of each approach, and will provide practical guidelines based in particular on the information gathered with the proposed V&V techniques. For example, in the claim-argument-evidence approach suggested by CEMSIS [2], V&V can be used to shape the claims and argument, and to provide the evidence part of the justification. Methods benchmarked in the BE-SECBS can be applied to the software reliability assessment [3].

3.2. Software reliability assessment case

Assessment of the reliability of safety-critical software in NPP includes several types of reasoning. Figure 2 illustrates the categories of claims needed to be present the "software reliability assessment case" in a justifiable manner. The purpose is to structure the issues related to the reliability assessment into smaller, manageable issues, which can be discussed as independently (or conditionally independently) of each other as reasonably possible.

Firstly, the purpose and scope of the software reliability assessment need to be defined, which — in simple terms — can be associated with the needs of PSA and its applications, e.g., to show the compliance with the numerical risk criteria for the NPP. Naturally, other purposes of software reliability assessment may be considered, but they are out of the scope of this paper. The needs of PSA can be presented by system failure

mode related claims which aim at defining the basic events for which reliability estimates are needed e.g. in PSA. For instance, in the case of a processing unit, the failure modes may be simplified into 1) loss of all functions (no output at all), 2) loss of one function (no actuation signal), 3) spurious signal. In addition to this, it is relevant to define the degree of CCF. The following CCF cases could be postulated: redundant units within a division, redundant units in redundant divisions, all units with same platform, and units with different platform.

A second category of reasoning is related to the properties of the software in order to limit the scope of residual software faults that are of concern. Cat. A software systems are designed following strict design principles and they go through a rigorous V&V process, which gives well-justified arguments to rule out a number of software fault types, e.g., software is designed to behave cyclically, behavior is time-based rather than event-based, and the operating system is designed not to be affected by the plant conditions.

The third category of claims concerns the actual assumptions used in the modeling of the software reliability once the scope of the assessment has been defined, i.e., the system failure mode and conceivable software fault modes are given.

The modelling approaches that can be used to support such cases are discussed in the sections below.

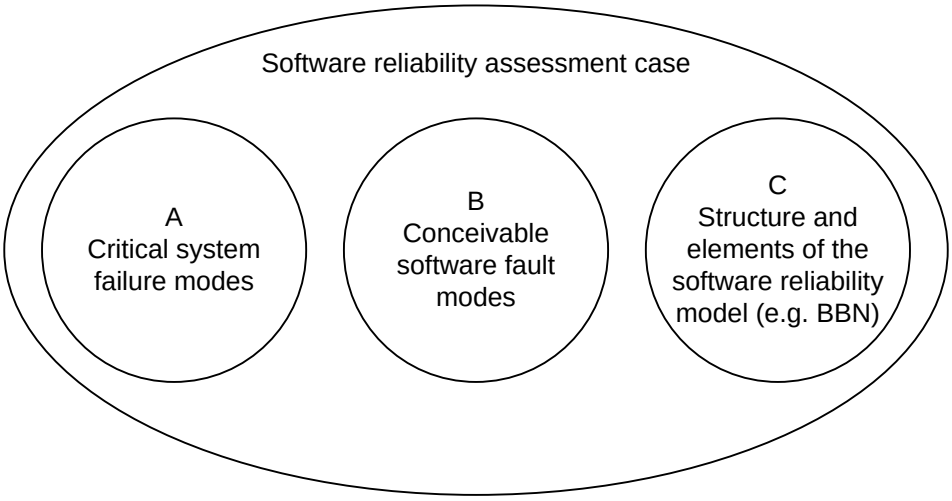

Figure 2. Categories of claims in a software reliability assessment case

4. ANALYTICAL APPROACH

The background of the proposed analytical method is the PSA. The method for developing digital system reliability estimates for use in a PSA may be decomposed into ten steps (see Table 1) [11].

The first five steps are performed by the PSA analyst and determine which digital systems are sufficiently important to the PSA to require the development of detailed system and software failure probabilities. This part provides the category A claims of Figure 2 (critical system failure modes).

The last five steps estimate the failure probabilities needed by the PSA and are performed by the I&C designer or engineer. This part provides the category B and C claims which are though closely interrelated.

Table 1. Analytical approach for developing digital system reliability estimates for PSA

Tasks performed by the PSA analyst
1. Identification of the functions, systems and components that are credited in the PSA and depend upon operation of software-based systems. Also, identification of those initiating events that could be triggered by particular software-related failures.
2. Identification of the effects that are desirable to avoid for functions, systems and components that are credited in the PSA, or the failure of which could trigger an initiating event.
3. Identification of the software-based systems, components and failure modes that contribute to the effects identified in Step 2.
4. Incorporating events in the PSA that represent the systems, components and failure modes identified in Step 3. Also, incorporating common-cause failures.
5. Sensitivity analysis to identify the failure modes of software-based functions and common cause failures that are critical to the PSA.
Tasks performed by the I&C engineer
6. Identification and classification of the failure mechanisms that can lead to the failure modes and software-related common-cause failure determined by Step 5.
7. Development of a reliability model of each software-based system identified in Step 6. A reliability model decomposes the system into "elements" such as individual computing units and data communication units. These can be further decomposed into major software elements such as operating system, standard library functions, and application-specific software. The model also represents the design features that could lead to common-cause failure of multiple elements, such as data communication, single-point vulnerabilities, identical or similar designs.
8. Identification and assessment of the various "defensive measures" taken to avoid, eliminate or tolerate certain types of errors, failure modes or failure mechanisms (including common-cause failure) that could affect elements of the digital systems reliability models.
9. Quantification of the rates of occurrence of the failure modes that could affect elements of the digital systems reliability models, and have not been rendered negligible by the defensive measures (see Figure 3).
10. Computation of the PRA critical parameters identified in Step 5 using the digital systems reliability models built in Step 6

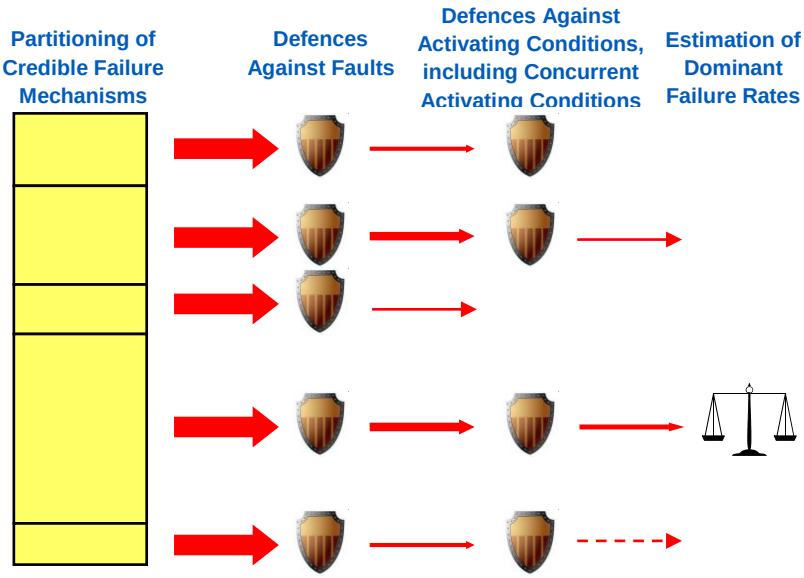

Figure 3. Use of defensive measures in the identification of the dominant failure mechanisms (steps 8 and 9)

In computing overall reliability, the proposed analytical approach takes account of the overall *system architecture*. For example, claims may be partitioned between software-based units and then combined to estimate the overall system reliability.

4. SOFTWARE RELIABILITY CLAIMS AND USE OF EVIDENCE

Software reliability can be measured in several manners. In HARMONICS, the following reliability metrics will be mainly considered:

- the probability that the software is imperfect (not fault-free): P(SW imperfect)
- the probability distribution for number of residual faults: P(N =n), N = number of faults
- the probability (or failure rate) of the critical digital system failure (due to a software fault): P(SW failure)
- the conditional probability of a common cause failure (called also beta-factor).

The two first ones are clearly interrelated, since P(SW imperfect) = P(N > 0), although there may be difference in which way evidence is used to assess them. While Fault-freeness is a true-false property to be assessed, P(N =n) requires a construction of a probability model for the number of residual faults.

The third and fourth metrics are needed for PSA. It may be difficult to directly estimate P(SW failure), and we may need to build the reasoning via the assessment of the imperfectness or number of faults. Estimation of beta-factor must be essentially based on the assessment of the software diversity.

Direct evidence for P(SW failure) can be measured directly using statistical testing to establish an upper bound on the failure rate. Evidence for the possible range of failure rates can also be obtained from past experience from a population of digital I&C systems [11]. For complex platform software, there could be multiple faults (so the probability of imperfection is close to one). However, the reliability estimate can make use of extensive operating experience that can be used to set an upper bound on the predicted failure rate [12].

Estimates for probability of imperfection can be established using a number of approaches:

- empirical evidence of residual faults in the software of similar nuclear control and protection systems [11]
- modeling the V&V development process to derive an estimate of the residual number of faults [13]
- measuring the complexity of the software unit.

The logic complexity estimation approach is more easily applied in cases where the code is auto-generated from high-level logic diagrams [14][15].

The analysis also takes account of *software diversity* where either:

- there are two or more software units that perform the same function
- there are software units that perform diverse functions that can prevent the same hazard.

For such cases, we need to estimate the probability of imperfection of the composite system from the probability of imperfection of individual units (e.g. SW1 and SW2). Clearly with a 1oo2 system function, both software units have to be imperfect for a failure to occur and ideally we would like to model this as a product of the imperfection probabilities of the two units. However, in practice, we will need to take account of factors affecting both units (such as a common requirements fault) that result in CCF. Estimation techniques of common cause failures can be based on empirical studies and subjective estimates (e.g. the likelihood of common fault).

To summarise, in any case some modeling is needed to derive the reliability metrics from the available evidence. When constructing the reliability model, the analysis needs to take account of uncertainty [16] in:

- the model parameters
- the model assumptions.

The model parameters are not point values, but cover a range of possible values that are more or less likely to occur. For example, the same statistical test evidence can be used to justify different reliability levels to different levels of confidence.

There is also uncertainty about the underlying model assumptions. For example, an assumption made in statistical testing is that the tests are representative of actual operation. The same assumption of representativeness has to be made when past operating experience from similar digital I&C systems is used as part of the justification, but the doubts are likely to be greater. To cope with uncertainty, the overall estimate needs to take account of the case where the assumption is wrong.

The reliability modeling approach will therefore use a combination of hard evidence and subjective estimates of probability ranges and doubts about underlying assumptions. It is an aim of HARMONICS to study how doubt on the modelling assumptions can be incorporated into the claimed reliability estimate.

Since the modeling of parameter ranges and assumptions doubts can become complex, one strand of our work will be the use of simplified, but conservative models with a limited number of point estimates (with confidence) that produce worst case failure rate estimates for use in a PSA [17].

The other strand of reliability modeling research is the use of Bayesian Belief Networks (BBNs) where we do not seek a worst case but a *distribution* of possible model parameters and model results. BBNs could for example be used to compute the distribution of P(SW imperfect) given information about the development process. BBNs can also be used at the overall argument level to combine evidence from testing and estimates of perfection [8]. The use of BBNs to support the models is discussed in more detail below.

5. BAYES BELIEF NETWORK APPROACH

Bayes belief network (BBN) is a general model for probabilistic inference so that the conditional dependences between the random variables are presented in a directed acyclic graph [18]. In the HARMONICS context, the random variables are reliability claims related to the software and various pieces of evidence available for reliability assessment. In particular, the analysis shall be based on the critical review of the evidence and analyses provided by the system vendor or the power utility applying for the license.

In the BE-SECBS project, BBN modelling had the following phases [3]:

1. Development of the map of evidence
    - identification of pieces of evidence to be included in the model
    - identification of the relationship between the pieces of evidence by engineering judgement
    - identification of the relationship between the evidence and the failure probability of the system
2. Definition the structure of the BBN model
    - definition of the variables (or the nodes) of the model; i.e. the definition of the variables measuring the degree of quality for evidence analysed in the tasks of the qualitative analysis
    - definition of the measurement of rating scales for each variable
    - definition of the probabilistic relationships and dependencies between the variables
3. Quantification of the model
    - quantification of the variable ratings by expert judgement
    - quantification of the needed probability distributions by expert judgement
    - propagation of uncertainties through the BBN model
4. Interpretation of the results.

The key issue is how different pieces of evidence are interpreted in a probability model context and how their interrelationships are assessed. Table 2 lists various sources of evidence available for the independent confidence building.

Table 2. Relevance of different pieces of evidence from the reliability estimation point of view

Source of evidence	Types of verification	Type of evidence on reliability
Process quality	Compliance with the requirements related to the process Experience of the developers Correctness of the development tools	Indirect evidence Fault freeness with certain confidence
Product analysis	Functional and structural properties of software Logic proof of correctness	Indirect evidence Fault freeness with certain confidence
Product testing	Validation of correct operation Confirmation of the reliability	Fault freeness with certain confidence Direct evidence given the representativeness of test cases

Indirect evidence needs to be measured by some rating scale which needs to be interpreted in terms of reliability. The use of indirect evidence in a justifiable manner would require some validation of the relationship between the evidence and reliability. Possibilities to validate indirect evidence for reliability estimation is one of the topics to be further investigated in HARMONICS.

Fault freeness evidence can be used to exclude certain software fault modes. From the reliability point of view there may remain doubt on the correctness of the evidence, which can be expressed as a probability. The way of expressing numerically (probabilistically) confidence on fault freeness evidence is one of the topics to be further investigated in HARMONICS. This is needed not only from the reliability assessment point of view but such an assessment is a relevant part of the overall safety justification of software systems.

Direct evidence is in principle the optimal case, but in reality the confidence on the representativeness of the data may need to be included in the assessment.

Figure 4 shows a principal belief network for the assessment of the software reliability. Various pieces of evidence can be used to estimate various reliability metrics (or bounds of them), as discussed above. Each link of the BBN refers to a particular reasoning problem for which a dedicated model needs to be developed — if that link is seen applicable. These will be studied in a series of case studies in HARMONICS.

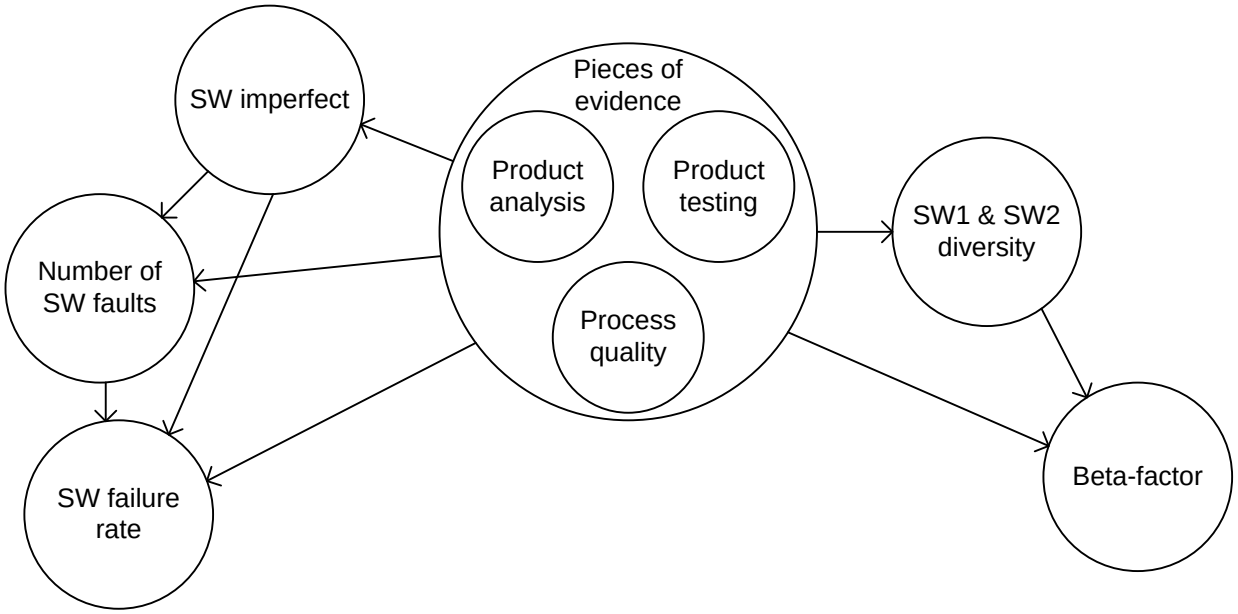

Figure 4. Principal belief network for the assessment of software reliability

6. SUMMARY AND FUTURE WORK

This paper has outlined the objectives of the HARMONICS project and some of the basic approaches that will be investigated for producing and justifying the reliability of software in a digital I&C system. Currently the project is evaluating the theory and tools to be used to construct the underlying models. These approaches will then be applied in a series of case studies using applications drawn primarily from digital protection systems. These include:

- estimating the reliability of a digital I&C platform
- empirical evidence of platform and application software reliability in past digital I&C systems
- estimating the probability of imperfection by modelling the V&V process
- estimating the reliability of a FPGA-based (field-programmable gate array) system.

We plan to report the estimation methods used and the results of achieved when the HARMONICS project is completed at the end of 2014.

Acknowledgements

HARMONICS is co-funded by the European Commission, the UK C&I Nuclear Industry Forum (CINIF) and the consortium organisations. The project consortium has five partners: VTT Technical Research Centre of Finland, Électricité de France (EDF), Institute for Safety Technology (ISTeC) from Germany, Adelard LLP from UK and Strålsäkerhetsmyndigheten (SSM) from Sweden. The public website address of the project is http://harmonics.vtt.fi. The authors also acknowledge the contributions of the other HARMONICS project members.

References

[1] Nuclear power plants. Instrumentation and control important to safety. Classification of instrumentation and control functions, IEC 61226, International Electrotechnical Commission, Geneva, ed. 3.0, 2009.

[2] CEMSIS. Cost Effective Modernisation of Systems Important to Safety. Work Package 0. Final Public Synthesis Report (first issue), 2004. http://www.cemsis.org/

[3] Kopustinskas, V., Kirchsteiger, C., Soubies, B., Daumas, F., Gassino, J., Péron, J.C., Régnier, P., Märtz, J., Baleanu, M., Miedl, H., Kersken, M., Pulkkinen, U., Koskela, M., Haapanen, P., Järvinen, M.L., Bock, H.W., Dreves, W. Benchmark Exercise of Safety Evaluation of Computer Based Systems (BE-SECBS Project). In Proc. of FISA-2003 conference, Luxembourg, November 10–13, 2003. ftp://ftp.cordis.europa.eu/pub/fp5-euratom/docs/fisa2003_2-8_be-secbs_en.pdf

[4] Licensing of safety critical software for nuclear reactors. Common Position of Seven European Nuclear Regulators and Authorised Technical Support Organisations, BEL V, Belgium, BfS, Germany, CSN, Spain, ISTec, Germany, NII, United Kingdom, SSM, Sweden, STUK, Finland, Revision 2010. http://www.hse.gov.uk/nuclear/software.pdf

[5] Chu, T.L., Martinez-Guridi, G., Yue, M., Lehner, J., Samanta, P. Traditional Probabilistic Risk Assessment Methods for Digital Systems, NUREG/CR-6962, U.S.NRC, Washington D.C., 2008, http://www.nrc.gov/reading-rm/doc-collections/nuregs/contract/cr6962/

[6] Function Safety of Electrical/Electronic/Programmable Safety-Related Systems, Parts 1–7, IEC 61508, various dates. International Electrotechnical Commission, Geneva.

[7] Haapanen, P., Helminen, A., Pulkkinen, U. Quantitative reliability assessment in the safety case of computer-based automation systems. STUK-YTO-TR 202, Radiation and nuclear safety authority, Helsinki 2004.

[8] Littlewood, B., Wright, D. The Use of Multilegged Arguments to Increase Confidence in Safety Claims for Software-Based Systems: A Study Based on a BBN Analysis of an Idealized Example. IEEE Trans. on Software Engineering, Vol. 33, No. 5, May 2007, 347–365.

[9] Chu, T.L., Yue, M., Martinez-Guridi, G., Lehner, J. Development of Quantitative Software Reliability Models for Digital Protection Systems of Nuclear Power Plants. NUREG/CR-7044, U.S.NRC, Washington D.C., 2011, Draft Report for Comment

[10] Safety Management Requirements for Defence Systems, Ministry of Defence Standard 00-56 Issue 4, June 2007.
[11] Estimating Failure Rates in Highly Reliable Digital Systems. EPRI TR 1021077, 2010. Limited distribution.
[12] Bishop P.G., Bloomfield R.E. A Conservative Theory for Long-Term Reliability Growth Prediction. IEEE Trans. Reliability, vol. 45, no. 4, Dec. 96, pp 550-560, ISSN 0018-9529
[13] Bloomfield R.E., Guerra A.S.L. Process modelling to support dependability arguments. In Proceedings of the International Conference on Dependable Systems and Networks, DSN 2002, Washington, DC, USA, June 2002.
[14] Bishop, P.G. MC/DC based estimation and detection of residual faults in PLC logic networks, ISSRE 2003, Fast Abstracts, Supplementary Proceedings, pp. 297-298, 17-20 November, Denver, Colorado, USA, 2003
[15] Bishop P.G. Estimating PLC logic program reliability. Safety Critical Systems Symposium, Birmingham, pp 179-193, Springer, 17th-19th February, 2004, ISBN 1-852-33800-8
[16] Bloomfield R.E., Littlewood, B. Confidence: its role in dependability cases for risk assessment. Proc International Conference on Dependable Systems and Networks (DSN2007) pp338-346, 2007
[17] Bishop P.G., Bloomfield R.E., Littlewood B., Povyakalo R., Wright D.R. Toward a Formalism for Conservative Claims about the Dependability of Software-Based Systems, IEEE Transactions on Software Engineering, pp 708-717, Vol. 37, No. 5, Sept/Oct 2011
[18] Pearl, J. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Representation and Reasoning Series (2nd printing ed.). San Francisco, California: Morgan Kaufmann. 2007.