A Method for Quantifying the Dependability Attributes of Software-Based Safety Critical Instrumentation and Control Systems in Nuclear Power Plants

Carol Smidts, Fuqun Huang, Xiang Li and Chetan Mutha
Department of Mechanical and Aerospace Engineering, The Ohio State University
Scott Laboratory W382, 201 W. 19th Ave. Columbus, OH 43210
smidts.1@osu.edu; huang.2315@osu.edu; li.984@buckeyemail.osu.edu; mutha.4@osu.edu

Ted Quinn
Technology Resources
23292 Pompeii Drive, Dana Point, CA. 92629
tedquinn@cox.net

ABSTRACT

The lack of systematic science-based methods for quantifying the dependability attributes in software-based instrumentation and control systems in safety critical applications has shown itself to be a significant inhibitor to the expanded use of modern digital technology in the nuclear industry. Dependability attributes include reliability, safety, availability, maintainability, and security (confidentiality and integrity). Modeling the dependencies between the dependability attributes is the first step towards dependability quantification. In this research we use two methods: structured expert opinion elicitation and (hierarchical) causal mapping to extract the dependencies. A panel of fourteen international experts was identified. Each expert filled a unique questionnaire, targeted towards dependability and attributes as per his/her expertise. The questionnaires were designed in a semi-structured format. The questions were designed to elicit the attributes encompassed by dependability, the root causes of each attribute, the dependencies between attributes, and how root causes and attributes affect dependability. Then the data from the expert elicitation was analyzed and converted to fourteen hierarchical causal maps. A hierarchical causal map is divided into three levels of detail: the top layer of the causal map is called the dependence level composed of the dependability attributes and interrelationships; the middle layer is called the Event of interest (EoI) level and expresses mechanisms leading to occurrence of the main event of interest (for instance a safety critical failure) for each dependability attribute; the third layer is called Measureable Concepts level, and is composed of measures for each of the EoI contributors. Finally, a merged causal map on the dependencies between dependability attributes was developed.

Key Words: Software Dependability, Nuclear Instrumentation and Control Systems, Causal Map, Reliability, Safety, Security, Availability, Maintainability

1 INTRODUCTION

Dependability is the ability to deliver service that can justifiably be trusted [1]. It is a widely accepted fact that challenges faced software dependability analysis due to the characteristics of software systems that are inherently different from hardware systems. The lack of systematic methods for quantifying the dependability attributes in software-based instrumentation and control systems has hindered the expanded use of modern digital technology in the nuclear industry [2, 3]. This issue is rendered significant by the fact that analog technology is aging and becoming obsolete (i.e. replacement parts are difficult or impossible to find), that the new generation of nuclear power plant engineers is now more familiar with digital technology than it is with analog technology, and that the benefits that digital technology offers cannot be tapped into [4]. These benefits include enhanced features, greater diagnostics, prognostics and on-line monitoring capabilities and added flexibility.

The current licensing methods and acceptance criteria for I&C systems in U.S. nuclear plants (new and current fleet) are based on NUREG-0800 Standard Review Plan (Ch 7: Instrumentation and Control), and the associated Branch Technical Positions BTP-7-14 (Guidance on Software Reviews for Digital Computer-Based Instrumentation and Control Systems) and BTP-7-19 (Diversity and Defense-in-Depth)[5]. The current regulatory review process described in BTP-7-14 does not use a quantitative basis and instead depends on the qualitative assessment of the reviewer. At the recent review of BTP-7-19 by the NRC Advisory Committee on Reactor Safeguards (ACRS), it was reaffirmed that no credit would be given for digital system reliability in the context of Digital Common Cause Failure (DCCF) review and approval, because the industry data was not sufficient to provide the NRC with a viable justification for accepting reliability-based analysis for digital system software.

To address the need for quantification and to give a more objective basis to the review process therefore reducing regulatory uncertainty, measures and methods are needed to assess dependability attributes early and throughout the life-cycle process of software development. Understanding the dependencies between dependability attributes is the initial step towards quantification. This paper presents the methods used to establish the dependencies between dependability attributes and our initial results.

2 EXPERT OPINION ELICITATION

Expert opinion elicitation is a systematic method used to extract and synthesize expert knowledge. This method is typically used when there is shortage of objective data or when data is unattainable. Expert opinion elicitation has been used in many scientific disciplines including risk analysis, decision under uncertainty, and to a certain extent in software engineering [7].

To elicit expert opinion, we first need to select domain experts, and then obtain their knowledge in an effective way. There are various ways to obtain expert knowledge, e.g., interviews and questionnaires. Questionnaires can be filled either on-line or off-line (e.g., sent by email). Off-line questionnaires allow experts to arrange their response time flexibly. Such advantages make off-line questionnaires especially suitable to gain access to experts who are busy and required to contribute to sophisticated problems. Therefore, we use questionnaires sent by emails to elicit expert opinion.

The number of experts need not be large because theoretically one perfect expert is sufficient. However, in practice even experts make mistakes, and hence using more than one expert is more appropriate. Another issue to be considered is the possible dependence between experts which leads to the diminishing return of adding extra experts [7]. Both issues were included in the design of the expert selection process

2.1 Expert Selection

Reference [7] provides five guidelines for expert selection, which can be summarized as follows:

1. Guideline 1- The experts should have demonstrated extensive experience in the related areas.
2. Guideline 2- Each expert should be versatile enough to address as many aspects of the problem as possible.
3. Guideline 3- The experts should represent a wide variety of experience as is obtained in academia, industry or government agencies.
4. Guideline 4- The experts should represent as wide a perspective of the issue as possible.
5. Guideline 5- The experts should be willing to be elicited under the methodology to be used.

Extensive experience as recommended in guideline 1 is assessed by an expert’s number of relevant publications for a specific attribute. Guideline 2 is satisfied by choosing experts that combine expertise in multiple attributes, i.e. relevant publications are identified for more than one attribute and experts should be versed in more than one attribute. Guideline 3 is not currently satisfied and will be addressed in future research. Guideline 4 is satisfied by verifying that the experts are not from the same institution or do not share publications. Satisfaction of guideline 5 is ensured by the content of the invitation email, which explicitly specifies that questionnaires will be used to collect opinions and used to build causal maps of dependencies between attributes that will be disseminated publically.

Figure 1 displays the procedure followed for expert identification and selection.

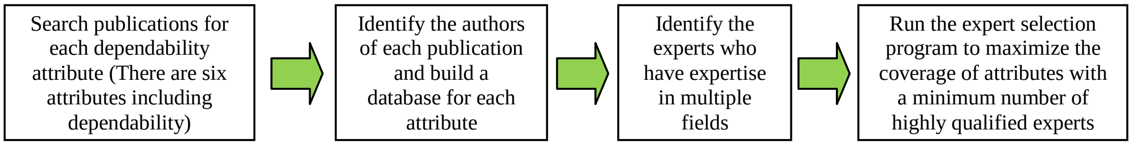

Search publications for each dependability attribute (There are six attributes including dependability)
Identify the authors of each publication and build a database for each attribute
Identify the experts who have expertise in multiple fields
Run the expert selection program to maximize the coverage of attributes with a minimum number of highly qualified experts

Figure 1. Procedure used for expert identification and selection

Google Scholar [8] was used from May 8th to May 27th 2014 to search publications on dependability attributes, from which an initial set of authors were identified. After authors were identified, their relevant publications were systematically searched and databases with <authors, relevant publications> were created for each attribute. Because there are six attributes (including dependability), there are $\binom{6}{2} = 15$ possible pairs of attributes of interest. To satisfy guidelines 1, 2 and 4, ideally, experts with at least knowledge on one pair are needed. To obtain such information, databases for each attribute are combined and authors with publications on pairs of attributes are identified. For example, suppose the virtual author John Smith has 6 publications on reliability, 3 publication on safety and 5 publications on dependability, then he is a potential expert on combinations of attributes <Reliability, Safety>, <Reliability, Dependability > and <Safety, Dependability>. His scores for each such pairs are hence 6*3=18, 6*5=30 and 3*5=15, respectively. Eventually we identified 24 potential experts who can cover at least one combination of attributes.

Once all the experts are identified, the last step is to maximize the experts' scores. More specifically, our objectives are: 1) to maximize the coverage of combinations of attributes; 2) to maximize the sum of the scores of each expert. The problem of selection falls in the category of knapsack problems and the computational complexity is NP-hard [9]. However, since the problem space is limited (i.e. the number of experts satisfying the selection criteria and number of experts required are close to each other), the computational time remained reasonable. Eventually 14 experts were selected and all combinations of interest except <Availability, Maintainability> were covered.

2.2 Questionnaire design

Questionnaire design is a multiple-stage process requiring iteration through testing. The information required to build a clear causal map for the dependencies between dependability attributes, the format, the sequence and layout of the questions have been considered in the design of the questionnaire. The issue of experts' bias has also been addressed.

2.2.1 Information required

The aim of the questionnaires is to identify the dependencies between dependability attributes, as well as the possible mechanisms that are at the source of the dependencies. Therefore, the questionnaires are designed to collect four categories of information: 1) the attributes that constitute dependability, 2) the relations between these attributes, 3) the root causes of the problems that may affect each attribute, and 4) how and under what circumstances the root causes become dependability problems.

We designed two types of questionnaires. The first one is aimed at the relations between two attributes, and the other focuses on the relation between one attribute and dependability. The questions and the information that we aim to gather for each question are shown in Table I.

Table I. A sample of the Questionnaires

Questionnaire Type	Questions		Aims of the questions	
Type 1: Paired attributes	Part I. Detailed Discussions	How do Attribute A and Attribute B Influence Each Other? I.1 *How does* Attribute A Influence Attribute B? I.2 *How does* Attribute B Influence Attribute A?	Obtain an overview of the relations between two attributes.	
	Part II. Identification of Dependence Mechanisms	II.1 What are the ***root causes*** of Attribute A problems?	Obtain the root causes of Attribute A problems.	Obtain the mechanisms underlying Attribute A problems
		II.2 What conditions, intermediate variables or events are required to transform the ***root causes*** into an Attribute A problem?	Obtain intermediate variables or events underlying Attribute A problems.	
		II.3 What are the ***root causes*** of Attribute B problems?	Obtain the root causes of Attribute B problems.	Obtain the mechanisms underlying Attribute B problems
		II.4 What conditions, intermediate variables or events are required to transform the ***root causes*** into an Attribute B problem?	Obtain intermediate variables or events underlying Attribute B problems.	
		II.5 What are the ***common causes*** for Attribute A problems and Attribute B problems?	Redundant questions that help the respondent differentiate the root causes of the two attributes and improve the confidence in the corresponding causal map.	
		II.6 What causes ***specifically*** lead to Attribute A problems, but do not lead to Attribute B problems?		
		II.7 What causes ***specifically*** lead to Attribute B problems, but do not lead to Attribute A problems?		
		II.8 What conditions, intermediate variables, states or events are required to transform an Attribute A problem into an Attribute B problem?	Obtain an explicit information on the scenarios under which Attribute A problems are transformed into Attribute B problems, and vice versa.	
		II.9 What conditions, intermediate variables, states or events are required to transform an Attribute B problem into a Attribute A problem?		
		II.10 The relation from Attribute A to Attribute B tends to be (multiple choices): ( ) a. Attribute A has positive effects on Attribute B, given the following assumptions/conditions/circumstances ________________________________. b. Attribute A has negative effects on Attribute B, given the following assumptions/conditions/circumstances ________________________________. c. No relation, they are independent.	Obtain explicit information on how Attribute A influences Attribute B.	
		II.11 The relation from Attribute B to Attribute A tends to be (multiple choices): ( ) a. Attribute B has positive effects on Attribute A, given the following assumptions/conditions/circumstances _________________________________. b. Attribute B has negative effects on Attribute A, given the following assumptions/conditions/circumstances _________________________________. c. No relation, they are independent.	Obtain explicit information on how Attribute B influences Attribute A.	
	Part III. Causal map	Please draw ***causal maps*** to illustrate how Attribute A and Attribute B affect each other based on the discussion and answers you provided to the questions on *Identification of Dependence Mechanisms (II)*	Obtain explicit causal map	
Type 2: Attribute-	Part I. Identification of Key	Please check the attributes (double click the appropriate box) that you think can **directly influence** *Software Dependability*	Identify the attributes that constitute dependability.	

dependability	Attributes	*Among* the attributes that you chose above, please identify the attributes that interact **directly** with Attribute C.	Identify the attributes that interact **directly** with Attribute C.
		Among the attributes you identified in Part I, Question 2, which attributes should be involved in a detailed discussion allowing you to meaningfully characterize the relationship between Attribute C and Dependability. This discussion will be the object of Part II.	Identify the attributes that interact **directly** with Attribute C and the expert is comfortable discussing.
		Please discuss the ***interaction mechanisms*** between Attribute C and the attributes identified in Part I, Question 3. Further, discuss how these attributes and their interactions ***influence Dependability***.	Obtain explicit information on how Attribute C-related attributes influence dependability.
	Part II. Identification of Dependence Mechanisms	Questions similar to Part II of the "Paired attributes" questionnaire.	Obtain information on the interaction mechanisms between the attributes.
	Part III. Causal map	Question similar to Part III of the "Paired attributes" questionnaire.	Obtain explicit causal map.

2.2.2 Response format

The response formats for questionnaires can be divided into three types: structured, unstructured and semi-structured [10]. Structured questionnaires consist of closed or prompted questions with predefined answers. The advantage of structured questionnaires is that they lend themselves easily to quantitative analysis. The disadvantage is that the researcher has to anticipate all possible answers with pre-coded responses. Unstructured questionnaires consist of open questions. The advantage of this type of questionnaire is that it supports exploration of new territories. The disadvantages are that: 1) the questionnaire may fail to gather the information that the researcher requires, 2) the data analysis may be challenging.

Our questionnaires are designed in semi-structured format to collect and explore as much information as possible for our specific aim. Semi-structured questionnaires are comprised of a mixture of closed and open questions. The use of semi-structured questionnaires enables collection of a mixture of qualitative and quantitative information. To ensure that we obtain the necessary variables for building the causal map, the questionnaire is "structured" at the high level. We have identified six types of concepts: root causes; intermediate variables; scenarios (assumptions/conditions/circumstances); dependability attributes and dependability. To collect a broad range of information, individual questions are designed as open questions.

2.2.3 Addressing expert bias

Due to the limits of human knowledge [11, 12] and human fallibility [13, 14], individual expert opinions are inevitably limited and may be biased. It is important to reduce expert bias during expert elicitation. The three typical biases [15] addressed in this research are as follows:

1) Absence of a link between two concepts in a causal map may not mean that the concepts are independent. Concepts that are separated in the map may actually be related, but the expert may not explicitly state the link in his/her interview response.

We have addressed this issue at two levels, the dependability level and the attribute level. At the dependability level ("attribute-dependability" questionnaire), we ask the experts to identify the attributes that directly interact with the attribute assigned to him/her. The answers to this question provide us evidence on the links between multiple attributes while the questionnaire remains focused on examining the relation between a single attribute and dependability. At the attribute level (Paired-attributes questionnaire), we have designed redundant questions (part I and part II) to support the drawing of the causal map (part III) and to help the experts identify the relations and explicitly represent them.

2) The way in which the expert words his/her answers may result in a reverse direction of the relationship between concepts in the causal map. A link from cause to effect may be represented as effect to cause. In this research, we have designed questions that are redundant to the explicit definition of the causal map. These questions help experts differentiate between causes and effects by explicitly identifying the root causes, intermediate variables and scenarios. This process can support the respondents in the construction of the cause-effects chain.

3) "A link between two concepts in the causal map implies that the relationship may either be direct or indirect. It is important to ensure that all the direct and indirect links between concepts are represented accurately in the causal map"[15]. We have introduced a question that requires the experts to provide intermediate variables, states or events that are required to transform problems with one attribute into problems with another attribute. Using the answers to this question, we can more accurately identify the relations as direct or indirect.

2.2.4 The sequence and layout of the questions

The ordering of the questions is important as it may stimulate logic reasoning. The general strategy is to ask easy and straightforward questions first and leave the more difficult or sensitive ones to a time at which the participants are primed [10].

The question "Detailed Discussion" is placed at the front of the questionnaire to initiate the respondent's free flow thinking on the overall relations between two attributes. The answers to this question are the most important as far as extracting the overall dependencies between attributes. Then questions on "Identification of Dependence Mechanisms" follow to awaken the respondent's causal thinking. The section titled "causal map drawing" is placed at the end of the questionnaire because it is the most difficult and is based on previous cognitive processes.

The layout of the questions is also important to improve the readability of the questionnaire. For each section, clear instructions are given. Sufficient space is provided for the answers. Larger amounts of space are left for the important open questions (e.g. "detailed discussion").

3 DATA ANALYSIS

Questionnaire Type 1 was sent to the experts whose expertise covered a pair of dependability attributes, and Questionnaire Type 2 was sent to the experts whose expertise covered one dependability attribute and dependability. Out of the 24 experts identified using the selection process specified in section 2.1, 19 were invited. Out of the 19 invitations, we received 14 responses in total. Since the questionnaires were designed to contain significant redundancies, each attribute was covered by multiple experts, shown in Table II.

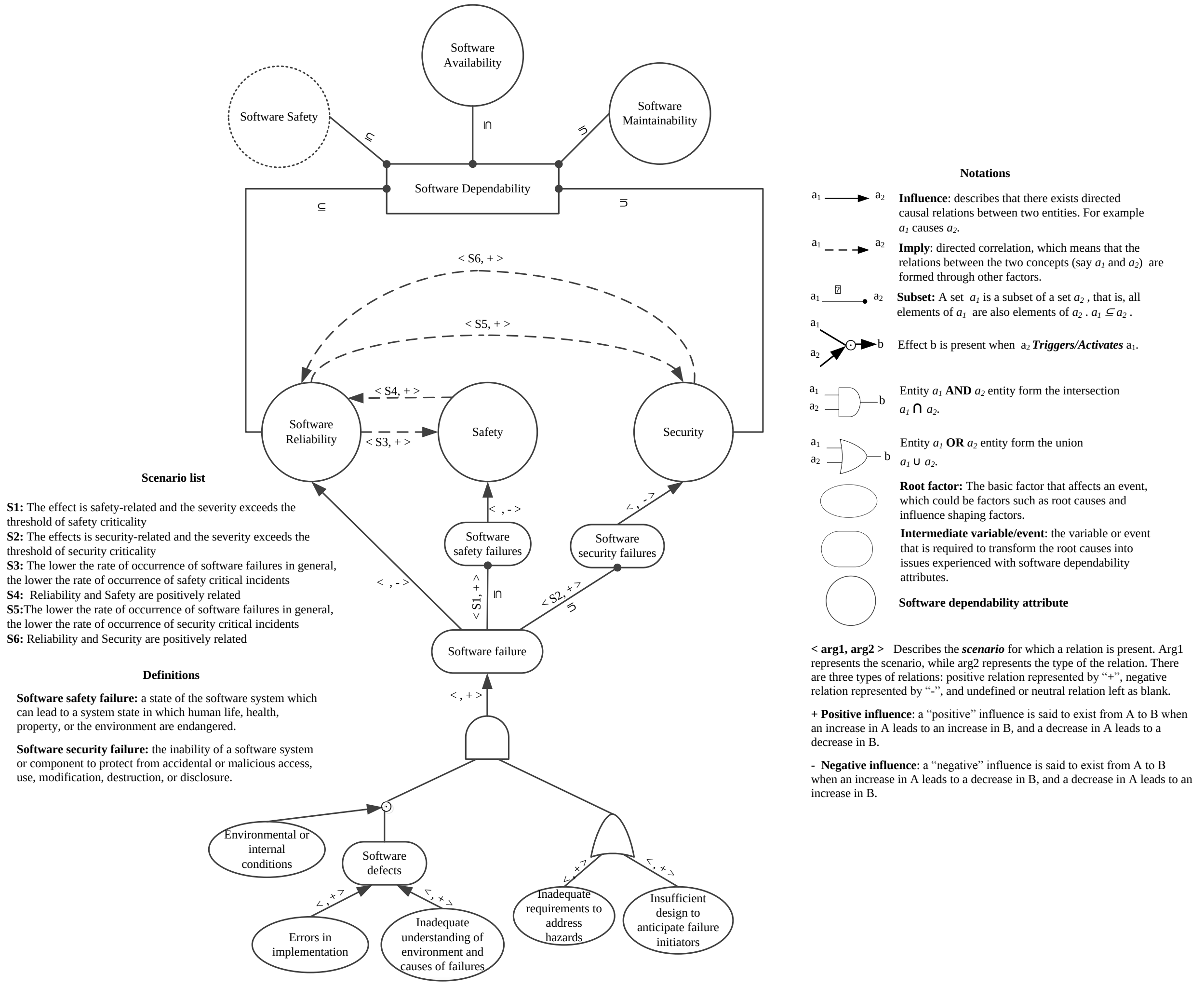
Table II. The coverage of software dependability and attributes

Software dependability and attributes	Covered times
Software dependability	8
Software reliability	7
Software safety	5
Software security	5
Software availability	5
Software maintainability	4

Experts provided no causal maps in the responses directly, but rather they provided text-based answers. We then use the causal map to represent the mental model embedded in the texts provided by the experts. A traditional causal map consists of three elements: nodes (used to represent the concept), edges (the direction of edges denoted as arrows implies believed causality) and ‘influence relationships’ (positive or negative) with strengths [16, 17].

There is only one type of relations existing in traditional causal maps: “influence”. There are no logic combinations between various variables. Furthermore, despite the fact that “time” is an essential dimension of causality, time sequencing is not clearly identifiable in conventional causal maps. There is also a lack of notations to capture the interaction mechanisms between variables. As a result, conventional causal maps are incapable of representing the complex causal mechanisms existing in software dependability domain. Therefore, we designed a set of new symbols to extract and represent the causal mechanisms and dependencies. A sample of the symbols and their interpretations are shown in Figure 2.

Each response was converted to an individual causal map, thus, we obtained 14 individual causal maps. An example individual causal map is shown in Figure 2.



Scenario list

S1: The effect is safety-related and the severity exceeds the threshold of safety criticality
S2: The effects is security-related and the severity exceeds the threshold of security criticality
S3: The lower the rate of occurrence of software failures in general, the lower the rate of occurrence of safety critical incidents
S4: Reliability and Safety are positively related
S5: The lower the rate of occurrence of software failures in general, the lower the rate of occurrence of security critical incidents
S6: Reliability and Security are positively related

Definitions

Software safety failure: a state of the software system which can lead to a system state in which human life, health, property, or the environment are endangered.

Software security failure: the inability of a software system or component to protect from accidental or malicious access, use, modification, destruction, or disclosure.

Notations

Influence: describes that there exists directed causal relations between two entities. For example a_1 causes a_2.

Imply: directed correlation, which means that the relations between the two concepts (say a_1 and a_2) are formed through other factors.

Subset: A set a_1 is a subset of a set a_2 , that is, all elements of a_1 are also elements of a_2 . $a_1 \subseteq a_2$.

Effect b is present when a_2 ***Triggers/Activates*** a_1.

Entity a_1 **AND** a_2 entity form the intersection $a_1 \cap a_2$.

Entity a_1 **OR** a_2 entity form the union $a_1 \cup a_2$.

Root factor: The basic factor that affects an event, which could be factors such as root causes and influence shaping factors.

Intermediate variable/event: the variable or event that is required to transform the root causes into issues experienced with software dependability attributes.

Software dependability attribute

< arg1, arg2 > Describes the ***scenario*** for which a relation is present. Arg1 represents the scenario, while arg2 represents the type of the relation. There are three types of relations: positive relation represented by “+”, negative relation represented by “-”, and undefined or neutral relation left as blank.

+ Positive influence: a “positive” influence is said to exist from A to B when an increase in A leads to an increase in B, and a decrease in A leads to a decrease in B.

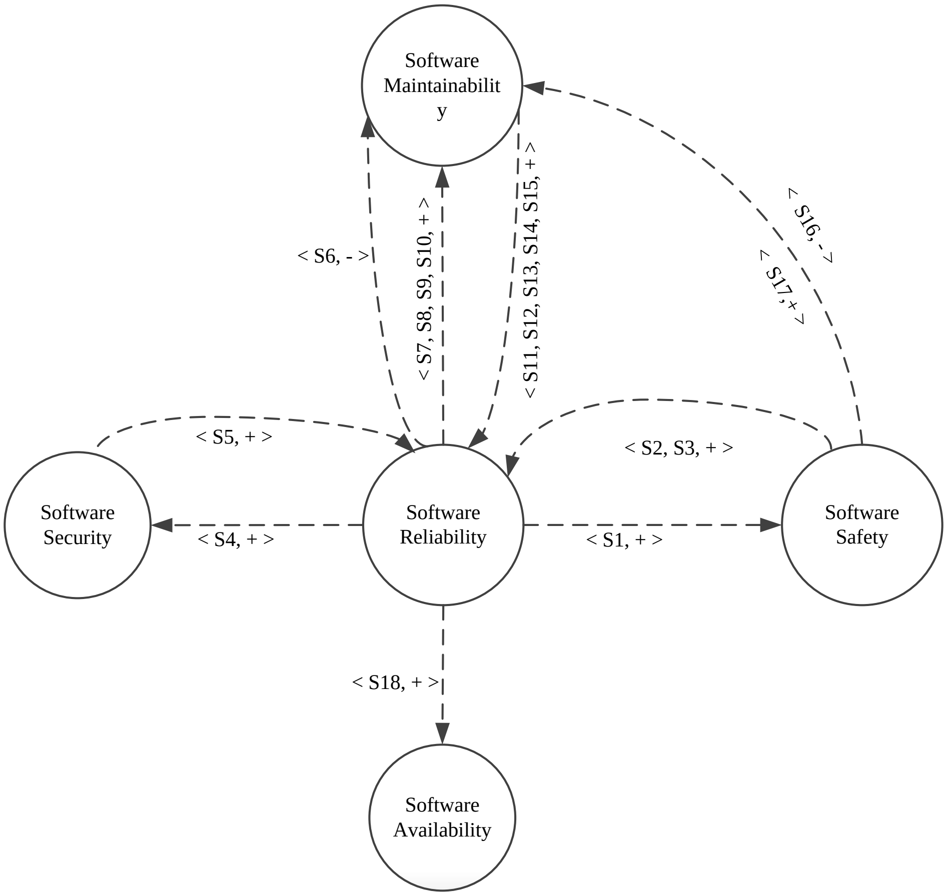
- Negative influence: a “negative” influence is said to exist from A to B when an increase in A leads to a decrease in B, and a decrease in A leads to an increase in B.

Figure 2. An example of individual causal maps

We merged these individual causal maps to obtain the aggregated knowledge on the dependencies between software dependability attributes. The results are shown in Figure 3.

From Figure 2 and Figure 3 we derive the following findings:

- Reliability, availability, maintainability, safety and security are subsets of dependability. There is no single measurement for dependability. Dependability measurements encompass measurements of these attributes. Different attributes may be emphasized for various contexts of application.
- Reliability, availability, maintainability, safety and security are related to each other, and under different scenarios/conditions, the relation can be either positive or negative. The relations between these attributes are correlations. The correlations between two attributes can be positive as well as negative, depending on the scenarios (contexts or assumptions).
- Correlations between dependability attributes are formed due to the existence of shared causal factors and mechanisms. Causal mechanisms are the key to explaining why correlations between two attributes are of different types under different scenarios. These causal mechanisms provide the fundamental basis to systematic measurements for dependability attributes.



Scenario list

S1: The lower the rate of occurrence of software failures in general, the lower the rate of occurrence of safety critical incidents.

S2: Reliability and Safety are positively related.

S3: When the system is not safe.

S4: The lower the rate of occurrence of software failures in general, the lower the rate of occurrence of security critical incidents.

S5: Reliability and Security are positively related.

S6: When Software Reliability is high due to high reliability requirement, which would increase the complexity of software.

S7: Lack of software documentation causes incomplete bug fix.

S8: Inadequate code patch causes incomplete bug fix.

S9: Unstable personnel is changed for corrective actions, causing potential Software Maintainability problems.

S10: The more reliable a system is the less time and money can be spent on fixing.

S11: Less maintainable software may make software maintainers produce wrong patches.

S12: When maintenance staff is unstable, correctness of maintenance actions may not be guaranteed, producing software faults which eventually causing software reliability problems.

S13: If maintenance actions change the software and violates the assumption on software operational environment, software may encounter failures, causing software reliability problems.

S14: If maintenance actions change the software fault tolerance design which becomes incorrect, then the defensive measure against faults may become ineffective, causing failures to occur.

S15: The required fix is made so the system is now reliable; the modification is done wrong and the system is now less reliable.

S16: The development capability remains constant and additional code complexity is added due to safety requirements.

S17: Assuming systems with significant safety requirements are developed by highly accomplished and experienced development teams.

S18: The restoration time and restoration probability are constant.

Figure 3. The dependencies between Software Dependability and Attributes

4 CONCLUSIONS

We have designed semi-structured questionnaires to elicit expert opinions in the software dependability domain. We have also defined a new causal mapping system to extract and represent expert knowledge more accurately. The initial results demonstrate that correlations exist between software dependability attributes, and the correlation types can vary due to the different assumptions made on their shared causal factors or mechanisms. These results provide the fundamental basis for software dependability measurements which we will investigate in the near future.

5 ACKNOWLEDGMENTS

This research is funded by the U.S. Department of Energy. We acknowledge the experts serving on the expert panel (in alphabetical order): Victor R. Basili, Norman Fenton, John de Grosbois, Myron Hecht, Erland Jonsson, Mohamed Kaâniche, Karama Kanoun, Michael R Lyu, Yashwant K. Malaiya, Daniel P. Siewiorek, Kishor S. Trivedi.

6 REFERENCES

1. Avizienis, J.C. Laprie, B. Randell, “Fundamental concepts of dependability,” University of Newcastle upon Tyne, Computing Science (2001).
2. C. Smidts, “Identification of Failure Modes of Software in Safety-Critical Digital I&C Systems in Nuclear Power Plants,” *Probabilistic Safety Assessment and Management Conference*, Helsinki, Finland, Jun. 25–29 (2012).
3. F. Brissaud, C. Smidts, A. Barros, and C. Bérenguer, “Dynamic Reliability of Digital-based Transmitters,” *Reliability Engineering & System Safety*, **vol. 96(7)**, pp. 793–813 (2011).
4. Smidts, C., Ming L. “NUREG/GR-0019: Software Engineering Measures for Predicting Software Reliability in Safety Critical Digital Systems,” USNRC, Washington DC, USA (2000)
5. “Standard Review Plan for the Review of Safety Analysis Reports for Nuclear Power Plants: LWR Edition--Instrumentation and Controls (NUREG-0800, Chapter 7),” USNRC, Washington DC, USA (2010).
6. Martin Neil, Bev Littlewood, Norman Fenton, “Applying Bayesian Belief Networks to System Dependability Assessment,” *Safety-Critical Systems: The Convergence of High Tech and Human Factors*, pp 71-94 (1996)
7. Li, Ming, and Carol Smidts, "A ranking of software engineering measures based on expert opinion." *IEEE Transactions on Software Engineering*, **vol. 29 (9)**, pp. 811-824 (2003).
8. “Google Scholar,” http://scholar.google.com/. Retrieve: Nov. 10, 2014 (2014)
9. Kellerer, Hans, Ulrich Pferschy, and David Pisinger, *Knapsack problems*. Springer (2004).

10. Hague, P., “Section 8: An Introduction to Questionnaire Design. In A Practical Guide to Market Research,” www.b2binternational.com/files/08-market-research- h8.pdfCachedSimilar (1988).

11. Fuqun Huang, Bin Liu, Bing Huang, “A Taxonomy System to Identify Human Error Causes for Software Defects,” Paper presented at the *The 18th international conference on reliability and quality in design*, Boston, http://www.mecheng.osu.edu/lab/risk/personnel/F_Huang.

12. Fuqun Huang, Bin Liu and Yichen Wang, “*Software psychology review*,” Computer Science, **vol. 40(3)**, pp. 1-7 (2013).

13. J. Reason, *Human Error*. Cambridge University Press, Cambridge, UK(1990).

14. Fuqun Huang, Bin Liu, You Song, Shreya Keyal, “*The links between human error diversity and software diversity: Implications for fault diversity seeking*,” Science of Computer Programming, **Volume 89, Part C**, pp.350-373 (2014).

15. Sucheta Nadkarni, “*Aggregated Causal Maps: An Approach To Elicit And Aggregate The Knowledge*,” Communications of the Association for Information Systems, **12**, pp.406-436 (2003)

16. Axelrod, R., *Structure of Decision: The Cognitive Maps of Political Elites*. Princeton University Press, Princeton, NJ (1976).

17. Eden C, “*Analyzing cognitive maps to help structure issues or problems*,” European Journal of Operational Research, **159**, pp. 673–686 (2004)