

Planning for Safety Evidence Collection: A Tool-Supported Approach Based on Modeling of Standards Compliance Information

Davide Falessi^{1,2}, Mehrdad Sabetzadeh¹, Lionel Briand^{1,3}, Emanuele Turella², Thierry Coq⁴ and
Rajwinder Panesar-Walawege^{1,3}

¹Simula Research Laboratory, Lysaker, Norway

²University of Rome Tor Vergata, Italy

³University of Oslo, Norway

⁴Det Norske Veritas, Paris, France

Abstract — *Safety-critical software-dependent systems such as those found in the avionics, automotive, maritime, and energy domains often need to be certified by a licensing or regulatory body based on one or more safety standards. Safety standards do not specify the details of the evidence that needs to be collected for the certification of a particular system because these standards need to be generalizable and applicable to a wide variety of systems. Without an upfront agreement between the system supplier and the certifier about the details of the evidence that needs to be collected, there will invariably be important omissions in the evidence information provided by the supplier, which will need to be remedied after the fact and at significant costs.*

The contributions of this article are twofold: we present both a flexible approach and a publicly available supporting tool for assisting suppliers and certifiers in developing an agreement about the evidence necessary to demonstrate compliance to a safety standard. The approach is model-based; specifically, the safety standard of interest is expressed via an information model. The supporting tool, which is available online, takes this information model as input and assists system suppliers and the certifiers in reaching a documented and consistent agreement about the safety evidence that needs to be collected.

Keywords- II.XIX.II.Planning for SQA and V&V, II.0.IV Standards, IV.II.15 Software and System Safety.



1 Introduction

Safety-critical software-dependent systems such as those found in the avionics, automotive, maritime, and energy domains often need to undergo a stringent certification process and be shown to comply with one or more safety standards. Our focus here is on managing the agreement about the evidence necessary to demonstrate compliance with the applicable safety standards. This is an important aspect of safety argumentation and assessment in practice [1].

The contributions of this article are twofold. We present both a model-based approach and a supporting tool to assist system suppliers and certifiers in elaborating, *ahead of time*, what safety evidence information should be collected. While our approach and tool are standard-independent, for clarity, we ground our discussions in this article on the IEC 61508 standard, which is one of most widely adopted generic standard for managing the functional safety of software-dependent systems [2].

2 Why develop an agreement about safety evidence?

Although safety standards provide some guidance for the collection of relevant safety information, this guidance is mostly textual, not expressed in a precise and structured form, and is hard to specialize for context-specific needs. Without an agreement between the supplier and certifier, there might be discrepancies between the ways in which they interpret the standards, giving rise to problems for both sides.

On the supplier side, there is a great risk that the information necessary for certification is not recorded during the development process and thus needs to be recovered after the fact. This can lead to significant cost overruns and delays in the deployment of the developed system. Indeed, given the difference between the times when the development and when the certification processes occur, the involved personnel may have moved to a different project, department, or company. Consequently, the necessary evidence may have been lost entirely or may need to be re-produced from scratch and often at extremely high costs. A high profile example of such problems occurred during the certification of the computer system on Airbus A400M where a change in certification requirements led to significant rework because the new certification requirements were misunderstood or neglected [3].

On the side of the certifier, the main problem caused by the absence of a clear agreement about the required evidence is that the documentation the certifier receives from the supplier lacks structure and might not be directly targeted at safety. Indeed, from our experience, because of the lack of an upfront

agreement, the suppliers often tend to provide large fragments of their existing documents with the hope that the certifier will find the required safety information. The result is that the certifier usually has to invest a significant amount of time and effort sifting through the provided documents, and in many cases not finding what they were looking for.

The motivation for our approach and the supporting tool is the lack of systematic support for suppliers and certifiers to negotiate and arrive at a consistent agreement about the evidence that should be collected and how this evidence should be delivered.

3 Overview of the questionnaire-based agreement process

3.1 A model-based agreement

An overview of our questionnaire-based agreement process is depicted in the center of Figure 1. The input is shown on the left of the figure, the actors on the top, and the output on the right. Briefly, the agreement process revolves around the notion of a *questionnaire*: the questions concern the details of the evidence to collect and the answers are the alternatives ways of recording and structuring the evidence.

The *administrator* defines the questionnaire for a given safety standard. The *supplier* proposes answers (i.e., possible specializations) and the *certifier* accepts or rejects the answers and provides the rationale for the decisions via comments. After the certifier agrees on the supplier's answers, the tool provides as output an *agreement document* (in PDF format) to review, print, and sign.

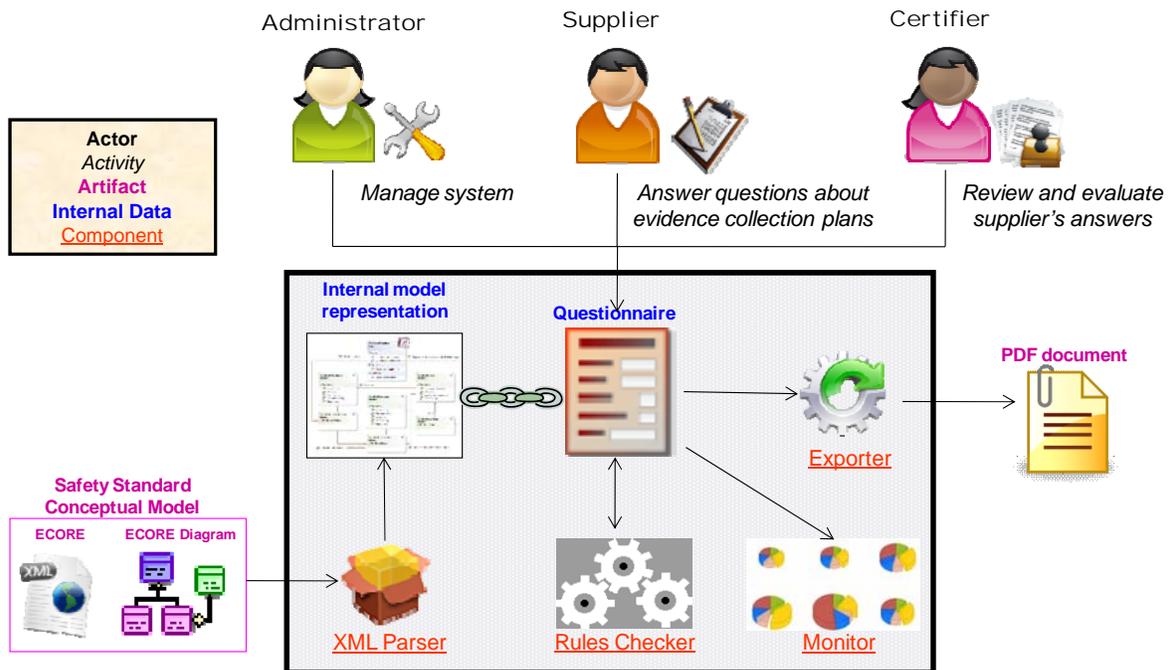


Figure 1: Overview of the solution for safety evidence planning.

Our approach to planning of safety evidence collection is *model-based*. Specifically, to manage the apparent complexity of safety standards and provide an explicit and precise interpretation of their content, we capture the core concepts of a given safety standard and relations among these concepts using an *information model*. In [4], we developed an information model for the IEC 61508 standard; it is a UML *class* model encoded in an Eclipse-compatible format (ECORE). We show in Figure 2 a fragment of this information model. Briefly, an agreement concerning this fragment must specify which techniques, and in particular which safety validation techniques, are carried out in which phases and by which agents in relation to the targeted safety integrity levels. In Section 4, we use the model fragment of Figure 2 for illustrating how a questionnaire can be built around the concepts and relations in an information model.

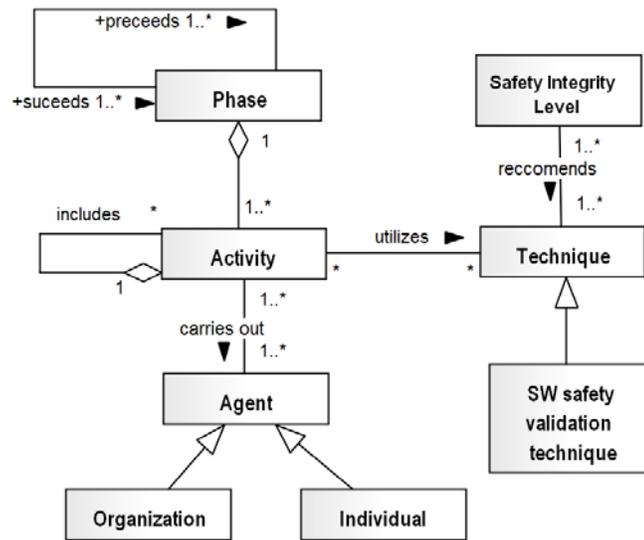


Figure 2: A fragment of the IEC 61508 information model used in this article for illustration

3.2 The EvidenceAgreement tool

We have developed a tool called EvidenceAgreement to support our approach. EvidenceAgreement is web-based and allows certifiers and suppliers to collaborate easily with one another even when they are located at different geographical sites. A key challenge was finding the right balance between formality and ease of use in order to increase the practical utility of the tool. In our implementation, we make use of the following technologies:

- The Java® programming language and the J2EE 5 platform.
- EclipseLink® for Object-Relational Mappings. EclipseLink provides an Eclipse-compatible implementation of the Java Persistence API.
- JAXB for generating Java code from an existing XML schema and binding XML data to Java objects.
- Vaadin® for creating rich internet applications using only Java. Vaadin is open-source and built on Google Web Toolkit (AJAX).

EvidenceAgreement is documented via UML diagrams and its documentation is available in both Enterprise Architect® and PDF formats. EvidenceAgreement, and its commented demo, are publicly available¹ at: <http://modelme.simula.no/evagr/index.html>.

4 The questionnaire

The administrator is in charge of creating a questionnaire for the standard to comply with. In practice, the administrator role is typically played by one or a group of experts (usually, certifiers) who can interpret the details of the relevant standards and enumerate alternative ways of achieving compliance in different contexts. In general, there is one information model per standard but several questionnaires can exist per information model. As a matter of good practice, we recommend the use of a single questionnaire per information model encompassing all the domains that the underlying standard applies to. However, while our experience indicates that this was possible with IEC 61508, we cannot be sure that having a single questionnaire can support any given standard in all possible domains. Therefore, the tool allows the association of multiple questionnaires to an information model.

The administrator assigns to the supplier and certifier a specific questionnaire to use to reach an agreement. Subsequently, the supplier and certifier need to authenticate and then choose the agreement to work on among the assigned ones. Both the supplier and the certifier can observe the status of the agreement, as illustrated in Figure 3. For each question, the tool shows the status in both text and color for easier comprehension. Pie charts are used to show the status of a package of the information model and the “Final status” aggregates the status information about the different types of questions. We describe the question types in the next section.

¹ To access the webpage, please use “evidence” and “simula11” as username and password, respectively. The password protection will be removed after the end of the review process.



Figure 3: Monitoring of the agreement completion status.

The administrator can assign to supplier and certifier an already made questionnaire or create new ones. To create a questionnaire the administrator should:

- Provide some basic information including questionnaire name, author, and brief description.
- Select an information model to use based on the standard that needs to be complied to. Any information model can be used as long as it is encoded in the E CORE format.
- Define the questions, answers, and their rules, as we are going to describe in the remainder of this section.

We note that, in our current implementation, we assume that the underlying information model does not evolve; hence, once the questionnaire has been created, no synchronization with the information model is required.

4.1 Questions Types

Questions about context: Different contexts may require different evidence. The contextual questions help the supplier better plan for evidence collection in a given context. A common contextual question is “In which domain will the product be deployed?” The answer to this question would affect the level of safety required. For example, the same fire monitoring and control system could be deployed in different domains – an offshore oil platform as opposed to an on-land refining facility. Each deployment would have different safety concerns and might need to comply with different safety levels.

Contextual questions are associated with the whole questionnaire, without any constraints on their number or the number of answers that can be given to each question. The supplier is supposed to answer the contextual questions at the beginning of the process, because the context has an overarching effect on all aspects of the evidence planning process. This type of questions is the only one that does not require an agreement on the certifier side; this because the context is fixed given the obligations that the supplier has to the end-customers. For the remaining types of questions, it is necessary that the certifier should review and agree (or disagree) with each of the answers provided by the supplier.

Questions about evidence concepts: These questions concern the classes in the information model and describe in textual form the types of evidence to specialize. The administrator creates a question of this type by selecting a given class of the information model. All classes of the information model must have one question of this type. An example of this kind of question, associated with the “software safety validation technique” class is: “Which are the adopted techniques for software safety validation?” Answers to questions of this kind describe the possible specializations of the evidence. For instance, the alternative answers for the previous question include “Probabilistic testing”, “Simulation and modeling”, and “Functional and black-box testing”. A questionnaire must have one question of this type for each class of the information model. The supplier can answer this type of questions by selecting among the pre-defined answers or by proposing new ones. Pre-defined answers are stored in the information model of a given questionnaire. To reach an agreement, the certifier must agree on all the answers of this type of questions and if necessary suggest additional answers.

Questions about relations between evidence concepts: These questions concern the relations between the classes in the information. After the supplier has provided answers to the questions about the classes (see above), s/he must elaborate on the relations between the classes. The questions for a given relation are automatically derived from the answers provided for the pair of classes linked by the relation.

The answer is of the open text type. For example, once the questions for “Agent” and “Software safety validation technique” types of evidence have been answered, the supplier can specify which agent will be in charge of applying which safety validation technique.

Questions about deliverables: It is important that the certifier and supplier agree on how the evidence should be delivered. Therefore, for each proposed evidence concept, the supplier needs to answer the question “In which deliverable(s) will you provide this type of evidence?” Deliverables include artifacts (e.g., a given type of documentation) and actions (e.g., a review meeting). In our tool, we populate the list of possible deliverables by DNV’s plan approval documentation types [5]. The supplier can choose among pre-defined answers or propose new ones. An agreement must have at least one agreed deliverable per evidence concept.

4.2 Rules and inconsistent states

Consistency, completeness and traceability among the provided answers are enforced by the use of *rules*. These rules are defined by the administrator and checked at run-time by the tool. The rules specify the constraints a questionnaire must meet. There are two types of rules:

- A *multiplicity rule* prescribes the minimum number of answers that the supplier must propose for a given question. For example, the standard may require that at least two different techniques (answers) are adopted for software safety validation.
- An *exclusion rule* avoids the coexistence of two specific answers, belonging to the same or different questions, i.e., the two answers cannot be both selected. For example, the answer “simulation and modeling” may be excluded when the answer “COTS (Commercial Off-The-Shelf) technology” has been selected.

The interaction among exclusion and multiplicity rules can lead to inconsistent states. For example, the supplier may not be able to meet the multiplicity constraint of one question because the available answers (for the current question) are excluded by answers to other questions. To illustrate, consider the following example. IEC 61508 specifies four Safety Integrity Levels (SILs), with SIL 1 being the lowest and SIL 4 the highest level. At SIL 4, the supplier often needs to choose at least two testing techniques (answers) for software safety validation. If two out of the three possible answers, e.g., “Probabilistic testing” and “Simulation and modeling” get excluded by answers to other questions, then there will be an inconsistency because there is no possible way in which two techniques for software safety validation could be proposed.

Inconsistent states require backtracking because the user needs to change the answers to one or more of the previously answered question(s) so that a feasible answer to the current question is no longer excluded. The tool supports the management of inconsistent states in two ways:

1. The tool prioritizes (ranks) the questions at run-time according to the likelihood of each question becoming infeasible to answer while respecting the multiplicity rules (see first column in Figure 3). The likelihood is estimated to be proportional to the number of required answers and the number of exclusion rules per answer. The user can thus proceed with answering the most constrained questions first, as per the order suggested by the tool.
2. For each alternative, the tool shows all the alternatives excluded by it, as exemplified in Figure 4. In this way, the supplier can avoid in advance to choose an answer that excludes a desired answer. The tool shows also the alternatives excluding it; in this way the supplier can easily realize which answer to choose to allow the current answer.

Question answering ✕

How do you verify software safety?

Answer	Y/N	Status	Rules
1 Probabilistic testing	<input type="checkbox"/>		->
2 Simulation and modelling	<input type="checkbox"/>		->
3 Functional and black-box testing	<input checked="" type="checkbox"/>	Agreed	->

Save and close
Close without saving

Required number of alternatives: minimum 0, maximum 3

Rules (click the button '->' near chosen alternative)

Direction	Trigger	Answer	Question
in	triggered	Yes	Do you use COTS?

Status changes history

Date	User	Role	Action	Comment
6 apr 2011 10:58:00	Davide Falessi	Certifier	3 Functional and black-box testing: Agree	
6 apr 2011 10:56:41	Emanuele Turella	Supplier	The user has selected the following alte	3 Functional and black-box testing

Figure 4: Example of excluded and excluding answers details. For a given answer of a question (A), the tool shows the related rules (B) and the log report (C). In (A), the tool shows the question (very top), the possible answers (first column), the chosen answers (ticked checkboxes), and the answers not chosen (empty checkboxes). Excluded answers are hidden. For a given answer, clicking on the “rules” button

will bring up (B). Specifically, in (B), we have one row per rule and the column “direction” describes if the exclusion is incoming/outgoing from/to another answer; the trigger column describes if the answer has been chosen or not. In this example, the answer “Simulation and modeling” is excluded (i.e. “incoming” rule) from an answer of the question “Do you use COTS?”. In (C), the tool reports the history of the agreement according to the date, the user, the role and the specific action performed on the question.

5 Agreement evolution

In our practical experience, after an agreement has been reached and the evidence has been collected and provided to the certifier, it is possible that, after the analysis of the provided evidence, the certifier may require further evidence. One frequent reason is that some aspects of the evidence may provide contradictory content or insufficient information, and additional information may be required by the certifier before making a decision. The agreement may therefore need to change after reviewing the collected evidence.

To update an already existing agreement, the certifier can change the status of the questions whose answers need to be revised from “Agreed” to “Partially agreed”, and provide the rationale behind this decision via comments. The supplier can then review these comments and propose modifications to their previous answers. For example, the “Probabilistic testing” and “Simulation and modeling” test results may turn out to be unsatisfactory, and the certifier may require further test results. The supplier can therefore propose to provide, say, “Functional and black-box” test results too.

Note that a revised agreement is subject to the same exclusion and multiplicity constraints as the original agreement. These constraints are automatically enforced as the agreement is being updated, thus ensuring that the agreement always remains in a consistent state.

6 Output

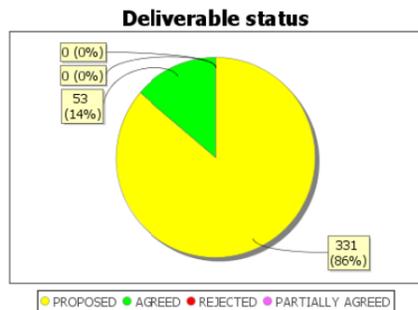
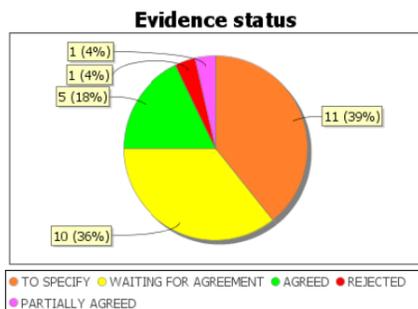
The tool generates as output a PDF document intended mainly as an appendix to the certification contract. The document is sketched in Figure 5. Specifically, the document can be customized to include the following elements:

- Pie charts
- Descriptions of questions and their status
- Description of context questions and their status

- Description of evidence types and their status
- Description of the mappings among evidence types and their status
- Descriptions of deliverables and their status
- A traceability matrix that maps evidence types to deliverables

Overall statistics

1. Statistics



Mapping of evidences to deliverables

1. Mapping matrix

	D1	D2	D3	D4	D5	D6	D7	D8
Q1_E1								
Q2_E1		X	X					
Q2_E2								
Q2_E3								
Q3_E1								
Q4_E1	X			X			X	
Q5_E1		X					X	
Q5_E2				X			X	
Q6_E1	X	X						
Q6_E2		X		X		X		
Q7_E1								
Q8_E1								
Q9_E1								
Q10_E1								
Q11_E1								
Q12_E1								
Q13_E1								
Q14_E1								

	Q1_E1	Q2_E1	Q2_E2	Q2_E3	Q3_E1	Q4_E1	Q5_E1	Q5_E2	Q6_E1	Q6_E2	Q7_E1	Q8_E1	Q9_E1	Q10_E1	Q11_E1	Q12_E1	Q13_E1	Q14_E1
D1																		
D2		X																
D3		X																
D4					X	X	X											
D5																		
D6									X									
D7					X	X	X											
D8																		

Textual description

1. Which design and coding standard have you adopted?

- Final status: To specify
- Evidence status: Waiting for agreement
- Deliverables status: To specify
- Proposed alternatives:
 - Use of coding standard
 - Proposed (03-nov-10 5:36 PM)
 - Final agreement: Proposed
 - Agreement on evidence: Waiting for agreement
 - Agreement on deliverables: Waiting for agreement
- History of evidence:
 - [Supplier] The user has selected the following alternatives:
 - Use of coding standard
- Specification of relations between concepts

2. Which techniques are you using for dynamic analysis and testing?

- Final status: Waiting for agreement
- Evidence status: Waiting for agreement
- Deliverables status: Partially agreed
- Proposed alternatives:
 - Test case execution from boundary value analysis
 - Proposed (22-lug-10 12:54 PM)
 - Final agreement: Proposed
 - Agreement on evidence: Agreed (undefined)
 - Agreement on deliverables: Waiting for agreement
 - History of deliverables:
 - [Certifier] Agreement set to: Proposed ()
 - [Certifier] Agreement set to: Agreed ()

Figure 5: An excerpt of a PDF document produced by EvidenceAgreement.

7 Conclusion

Our approach, which is based on constructing and specializing information models for safety standards, allows certifiers and suppliers to develop a negotiated and structured agreement about the evidence necessary for compliance to these safety standards. Though the tool has not been formally evaluated, we note that it is *not* a major (and disruptive) break from current practice, but instead a more effective way to do what is being done already through the manual completion of a plethora of different

checklists and spreadsheets. Our approach aims at systematizing the specification, management, and completion of these checklists and spreadsheets, and further providing automated support for maintaining the consistency of the provided answers, status monitoring, and report generation.

The sheer size and complexity of the safety evidence for large systems has been a constant challenge for safety engineers. Lewis [6], and Cockram and Lockwood [7] suggest the construction of *electronic* safety cases, so that one can dynamically query the safety information, instead of having to go through large physical documents. In the future, we plan to derive from the agreements generated by our approach data schemas for constructing and managing safety case databases that can be automatically analyzed.

8 Acknowledgement

This work was conducted as part of the ModelME! Project, which is a joint research effort between Det Norske Veritas (DNV) and Simula Research Laboratory.

Related work (sidebar)

Safety certification is one facet of the more general problem of compliance management [8], whose scope further encompasses topics such as process, medical, and environment regulations. A wide array of techniques and commercial tools already exist to this end, enabling the execution and monitoring of compliance-related activities. A related notion to compliance management is that of service-level management [9], aimed at developing a formal agreement for rendering of services and ensuring that the agreed services are duly delivered. Our present work could serve as an input to the existing compliance and service-level management tools (e.g., among others, IBS's CompliantPro² and MetricStream's compliance management software³) which focus on the concrete collection and validation of evidence. In this context, the main contribution of our work is the use of information models for formalizing the interpretation of safety standards and utilizing these models for guiding decisions about what evidence to collect.

Denney and Fischer [10] propose the idea of a Software Certificate Management (SCM) system to keep track of the links between safety evidence and the certificates issued based on the evidence. Their goal is to enable engineers to verify the validity of the certificates as the system and the evidence evolve.

² <http://www.ibs-us.com/en/products/compliantpro>

³ <http://www.metricstream.com/>

However, the work is not aimed at specifying the details of the safety evidence that needs to be collected, which was the subject of our work in this article.

Making safety evidence collection more systematic has been noted as an important problem in the literature before. In particular, Lewis [6] highlights the need for having a structured web of safety information covering not only the hazards and safety requirements but also, among others, the development process, hardware elements, human agents, and verification and validation results. Compliance assessment schemes such as CASS [11] for IEC 61508 partially address this problem by establishing guidelines for recording conformity. However, these schemes are still at a high level of abstraction and need to be further specialized for a given domain or system. Our approach addresses this gap by helping with the specialization of safety information according to the needs of a particular context.

Our work relates most closely with questionnaire-based elicitation techniques [12]. What differentiates our work is the use of model-driven engineering concepts to facilitate the specification of questions and possible answers, for ensuring coverage of the underlying safety standards, and for maintaining consistency between the provided answers.

AUTHOR BIOGRAPHIES

Davide Falessi is an adjunct research scientist in the Software Engineering Department at Simula Research Laboratory (Norway) and an adjunct lecturer at DISP at University of Rome “TorVergata” (DISP). His main research interest is in devising and empirically assessing scalable solutions for the development of complex software-intensive systems with a particular emphasis on architecture, requirements, and quality. Falessi has a PhD and an MSc summa cum laude in Computer Engineering from the University of Rome “TorVergata”. Contact him at d.falessi@ieee.org.

Mehrdad Sabetzadeh is a research scientist in the Software Engineering Department at Simula Research Laboratory. His main research interest is model-based development with an emphasis on modeling, analysis, and certification of safety-critical embedded software. Sabetzadeh received a PhD and an MSc in Computer Science both from the University of Toronto. He is a member of the IEEE Computer Society. Contact him at mehrdad@simula.no.

Lionel Briand is heading the Certus center for software verification and validation at Simula Research Laboratory and is a professor in the Department of Informatics at the University of Oslo. His research interests include model-driven development, testing and quality assurance, and empirical

software engineering. Lionel is currently the co-editor-in-chief of the Empirical Software Engineering Journal (Springer) and serves in the editorial boards of the Software and Systems Modeling (Springer) and Software Testing, Verification, and Reliability (Wiley) Journals. He has been on the program, steering, or organization committees of many international conferences. Briand received a PhD in computer science, with high honors, from the University of Paris XI, France. He is a fellow member of the IEEE. Contact him at briand@simula.no.

Emanuele Turella is a software architect at the National Institute of Nuclear Physics in Rome, Italy. He recently obtained the Masters Degree summa cum laude from the University of Rome, Tor Vergata, with a thesis based on the EvidenceAgreement tool. Contact him at eturella@gmail.com.

Thierry Coq is Project Director of Maritime & Energy at DNV IT Global Services. He has over twenty years of industry experience in the development and certification of software-dependant systems including pharmaceutical distribution, aeronautics, aerospace, automotive, and programmable electronic systems. He obtained a Masters degree from Ecole centrale de Paris. Contact him at thierry.coq@dnv.com.

Rajwinder Kaur Panesar-Walawege is a PhD candidate in the Software Engineering Department at Simula Research Laboratory and in the Department of Informatics at University of Oslo. She has over seven years of industry experience in the development of safety-critical systems. Her main research interest is model-based software development and software safety management. Panesar-Walawege has an MSc in Computer Science from the University of Victoria, Canada. Contact her at rpanesar@simula.no.

9 References

- [1] T. P. Kelly, "Arguing Safety - A Systematic Approach to Managing Safety Cases," University of York, 1998.
- [2] "Functional Safety of Electrical / Electronic / Programmable Electronic Safety-related Systems (IEC 61508)," *International Electrotechnical Commission*, 2005.
- [3] "<http://www.defensenews.com/story.php?i=4078604>," *Airbus A400M* 2009.
- [4] R. K. Panesar-Walawege, *et al.*, "Characterizing the Chain of Evidence for Software Safety Cases: A Conceptual Model Based on the IEC 61508 Standard," in *Proceedings of the 2010 Third International Conference on Software Testing, Verification and Validation*, 2010, pp. 335-344.
- [5] D. N. V., "RECOMMENDED PRACTICE DNV-RP-A201 - Plan Approval Documentation Types," <http://exchange.dnv.com/publishing/Codes/download.asp?url=2010-04/rp-a201.pdf>, 2010.
- [6] R. Lewis, "Safety Case Development as an Information Model," in *Seventeenth Safety-Critical Systems Symposium - Safety-Critical Systems: Problems, Process and Practice*, 2009.

- [7] T. Cockram and B. Lockwood, "Electronic Safety Case: Challenges and Opportunities," in *Safety-Critical Systems, Current Issues, techniques and standards*, F. Redmill and T. Anderson, Eds., ed, 2003.
- [8] M. Silverman, *Compliance Management for Public, Private, or Non-Profit Organizations.* : McGraw-Hill, 2008.
- [9] R. Sturm and W. Morris, *Foundations of Service Level Management*: Sams, 2000.
- [10] E. Denney and B. Fischer, "Software Certification and Software Certificate Management Systems," presented at the ASE'05 Workshop on Software Certificate Management, 2005.
- [11] CASS. Accredited Certification for Safety Systems to IEC 61508 and Related Standards. <http://www.61508.org/cass.htm>.
- [12] W. Foddy, *Constructing questions for interviews and questionnaires.*: Cambridge University Press, 1994.