Information and Software Technology xxx (2014) xxx–xxx

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

An extended systematic literature review on provision of evidence for safety certification

Sunil Nair [a,*], Jose Luis de la Vara [a], Mehrdad Sabetzadeh [b], Lionel Briand [b]

[a] Certus Centre for Software V&V, Simula Research Laboratory, P.O. Box 134, 1325 Lysaker, Norway
[b] SnT Centre for Security, Reliability and Trust, 4 rue Alphonse Weicker, L-2721 Luxembourg, Luxembourg

ARTICLE INFO

Article history:
Received 27 August 2013
Received in revised form 6 March 2014
Accepted 9 March 2014
Available online xxxx

Keywords:
Safety–critical systems
Safety standards
Safety compliance
Safety certification
Safety evidence
Systematic literature review

ABSTRACT

Context: Critical systems in domains such as aviation, railway, and automotive are often subject to a formal process of safety certification. The goal of this process is to ensure that these systems will operate safely without posing undue risks to the user, the public, or the environment. Safety is typically ensured via complying with safety standards. Demonstrating compliance to these standards involves providing evidence to show that the safety criteria of the standards are met.

Objective: In order to cope with the complexity of large critical systems and subsequently the plethora of evidence information required for achieving compliance, safety professionals need in-depth knowledge to assist them in classifying different types of evidence, and in structuring and assessing the evidence. This paper is a step towards developing such a body of knowledge that is derived from a large-scale empirically rigorous literature review.

Method: We use a Systematic Literature Review (SLR) as the basis for our work. The SLR builds on 218 peer-reviewed studies, selected through a multi-stage process, from 4963 studies published between 1990 and 2012.

Results: We develop a taxonomy that classifies the information and artefacts considered as evidence for safety. We review the existing techniques for safety evidence structuring and assessment, and further study the relevant challenges that have been the target of investigation in the academic literature. We analyse commonalities in the results among different application domains and discuss implications of the results for both research and practice.

Conclusion: The paper is, to our knowledge, the largest existing study on the topic of safety evidence. The results are particularly relevant to practitioners seeking a better grasp on evidence requirements as well as to researchers in the area of system safety. As a major finding of the review, the results strongly suggest the need for more practitioner-oriented and industry-driven empirical studies in the area of safety certification.

Abbreviations: AADL, Architecture Analysis & Design Language; ACRuDA, Assessment and Certification Rules for Digital Architectures; ASA, Automated and Structured Analysis; ASCAD, Adelard Safety Case Development; BBN, Bayesian Belief Networks; CAE, Claims, Arguments and Evidence; CCS, Calculus of Communicating Systems; CDL, Configuration Deviation List; CENELEC, Comité Européen de Normalisation Electrotechnique (European Committee for Electrotechnical Standardization) ; CMA, Common Mode Analysis; COTS, Commercial Off-The-Shelf; CSP, Communicating Sequential Processes; DECOS, Dependable Embedded COmponents and Systems; DOVE, Design Oriented Verification and Evaluation; ECHA, Environmental Condition Hazard Assessment; EMFI, Electromagnetic Fault Injection; ETA, Event Tree Analysis; EVA, Evidence Volume Approach; FFA, Functional Failure Analysis; FFPA, Functional Failure Patch Analysis; FHA, Functional Hazard Analysis; FMEA, Failure Mode, Effects Analysis; FMECA, Failure Mode, Effects and Criticality Analysis; FMEDA, Failure Modes, Effects and Diagnostic Coverage Analysis; FMES, Failure Mode and Effect Summary; FPGA, Field-programmable gate array; FPTC, Fault Propagation and Transformation Calculus; FPTN, Failure Propagation and Transformation Notation; FSM, Functional Safety Management; FTA, Fault Tree Analysis; GQM, Goal Question Metric; GSN, Goal Structuring Notation; HAZID, Hazard Identification Study; HAZOP, HAZard and Operability; HEP, Human Error Prediction; HHA, Human Hazard Analysis; HOL, Higher Order Logic; HRA, Human Reliability Analysis; IEC, International Electro-technical Commission; IET, Institution of Engineering and Technology; IHA, Intrinsic Hazard Analysis; ISO, International Organization for Standardization; MDE, Model-Driven Engineering; MC/DC, Modified Condition/Decision Coverage; MMEL, Master Minimum Equipment List; MTBF, Mean Time Between Failures; MTTF, Mean Time To Failure; OCL, Object Constraint Language; OS, Operating System; PHA, Preliminary Hazard Analysis; PRA, Particular Risk Analysis; PS, Primary Study; PSAC, Plan for Software Aspects of Certification; QA, Quality Assurance; RASP, Risk Assessment of Structural Part; RTCA, Radio Technical Commission for Aeronautics; RTOS, Real-Time OS; SACM, Structured Assurance Case Metamodel; SAL, Safety Assurance Level; SAS, Software Accomplishment Summary; SCMP, Software Configuration Management Plan; SDP, Software Development Plan; SEAL, Safety Evidence Assurance Level; SHARD, Software Hazard Analysis and Resolution in Design; SIL, Safety Integrity Level; SLR, Systematic Literature Review; SQA, Software QA; SRS, Software Requirements Specification; SSG, Safety Specification Graph; SVP, Software Verification Plan; SWIFI, Software Implemented Fault Injection; TPTP, Thousands of Problems for Theorem Provers; UAS, Unmanned Autonomous Systems; V&V, Verification and Validation.

* Corresponding author. Tel.: +47 40 64 40 46; fax: +47 67 82 82 01.
E-mail addresses: sunil@simula.no (S. Nair), jdelavara@simula.no (J.L. de la Vara), mehrdad.sabetzadeh@uni.lu (M. Sabetzadeh), lionel.briand@uni.lu (L. Briand).

http://dx.doi.org/10.1016/j.infsof.2014.03.001
0950-5849/© 2014 Published by Elsevier B.V.

Contents

1. Introduction

A safety–critical system is one whose failure may cause death or injury to people, harm to the environment, or substantial economic loss [5]. In domains such as aviation, railway, and automotive, such systems are typically subject to a rigorous safety assessment process. A common type of assessment, usually conducted by a licensing or regulatory body, is *safety certification*. The goal of safety certification is to provide a formal assurance that a system will function safely in the presence of known hazards [PS93]. Safety certification can be associated with the assessment of *products, processes, or personnel*. For software-intensive safety–critical systems, certification of products and processes are regarded as being the most challenging [PS93].

Assessing and assuring safety of a system relies on building sufficient confidence in the safe operation of the system in its operating context. This confidence is often developed by satisfying safety objectives that mitigate the potential safety risks that a system can pose during its lifecycle. The safety objectives are usually established by a set of industry-accepted criteria, typically available as standards. Examples of safety standards include IEC61508 [11] for a broad class of programmable electronic systems, DO-178C [7] for aviation, the CENELEC standards (e.g., [33]) for railway, and ISO26262 [8] for the automotive sector.

Demonstrating compliance with safety standards involves collecting evidence that shows that the relevant safety criteria in the standards are met [16]. Although, safety standards prescribe the procedures for compliance, it often proves to be a very challenging task to the system suppliers due to the fact that these standards are presented in very large textual documents that are subject to interpretation. In general, evidence can be defined as *"The available body of facts or information indicating whether a belief or proposition is true or valid"* [30]. For realistically large systems, however, one can seldom argue that evidence serves as a definitive proof of the truth or validity of safety claims, but only whether the evidence is sufficient for building (adequate) confidence in the claims. Hence, we define evidence for safety certification as *"information or artefacts that contribute to developing confidence in the safe operation of a system and to showing the fulfilment of the requirements of one or more safety standards"*. Some generic examples of safety evidence are test results, system specifications, and personnel competence.

The lack of consistent interpretation of a standard can lead to misunderstanding the evidence needs. Failing to clearly understand the evidence needs for assessing a system can result in two main problems [34,PS145]. First, the supplier may fail to record critical details during system development that the certifier will require later on. Building the missing evidence after-the-fact can be both expensive and laborious. Second, not knowing ahead of time what the certifiers will receive as evidence may affect the planning and organisation of the certification activities. In particular, the certifier may find it hard to develop sufficient confidence in the system undergoing certification if the evidence requirements have not been negotiated and agreed with the supplier a priori [PS54,15].

Apart from understanding and precisely defining the evidence requirements, attention needs to be paid to how this evidence is organised and assessed for adequacy. If the evidence is not structured properly, its sheer volume and complexity can jeopardize the clarity of the safety arguments [PS124]. Furthermore, it is important to be able to determine how definitive and credible the evidence is. Though safety standards mandate adequate evidence to show compliance, they are vague on what adequate means in a particular context, often intentionally and for the sake of being general.

The main objective of this paper is to synthesise the existing knowledge in the academic literature about safety evidence, concentrating on the three facets outlined above: the information that constitutes evidence; structuring of evidence; and evidence assessment. The term *evidence provision* is used hereafter to collectively refer to these three facets. Alongside, we analyse the challenges and needs in safety evidence provision and perform a domain analysis [15] to identify the commonalities among different application domains for this purpose.

We achieve our objective by means of a *Systematic Literature Review* (SLR) – a documented and repeatable process through which the literature on a given subject is examined and the current state of knowledge is recorded [18]. The main advantage of a SLR, when compared to ad hoc search, is that it provides a higher degree of confidence about covering the relevant literature and thus minimises subjectivity and bias.

Our SLR draws on 218 peer-reviewed publications, selected out of 4873, through a multi-stage process. A key feature of the review is that it does not restrict itself to a particular domain or safety standard. This broad scope in the search gives us deeper insights on the state of the art. Additionally, the breadth helps in understanding the commonalities among the different domains in terms of how evidence is perceived, structured and assessed, in turn enabling improvements in the domains that do not yet enforce stringent certification requirements, e.g., the automotive sector.

As part of our work, we classify into a hierarchical taxonomy the various information and artefacts considered as evidence for compliance with safety standards. The taxonomy includes 49 basic evidence types and is, to our knowledge, the most comprehensive classification of safety evidence built to date. This taxonomy is a good reference for understanding and further elaborating the evidence requirements for specific standards and specific systems. The other outcomes of the SLR, namely the survey of approaches for evidence structuring and assessment, the overview of challenges and needs, and a domain analysis to identify commonalities, will be a useful guide for developing a detailed map of the field and for defining a future research agenda on safety certification. Our study notably indicates that a large majority of the approaches surveyed have not been validated in realistic settings and thus provide little information about their practical utility. An important recommendation for future research on safety certification is therefore for the research to be more rigorous from an empirical standpoint and more oriented towards industry needs.

The SLR has been conducted as part of OPENCOSS [25], which is a large-scale European research project on safety certification in the railway, aviation and automotive domains. The work we present here extends an earlier conference paper [21]. The main extensions are: (1) the addition of a new data source, namely Google Scholar, thereby increasing the number of primary studies; (2) significant expansion of the description of the research method and the results; and (3) our domain analysis (mentioned above).

The rest of the paper is organised as follows. Section 2 discusses related work. Section 3 describes the research method used. Section 4 presents the SLR results. Section 5 discusses the implications of these results on research and practice. Section 6 discusses the threats to validity of the review. Finally, Section 7 presents our conclusions and future work.

2. Related work

Several papers discuss the notion of evidence in specific situations and how this evidence can be structured and assessed. We do not treat these as related work but rather as the primary studies for our SLR. The discussions in this section are therefore targeted at contrasting our work with the more generic classifications of safety evidence as well as the relevant existing SLRs.

Some threads in previous work, e.g. [PS121], address the problem of safety evidence classification through focusing on safety standards such as IEC61508. Further threads, e.g. [17], consider the structuring of evidence for safety cases. A *safety case* is a structured argument aimed at providing a compelling, comprehensive, and valid case that a system is safe for a given application in a given operating environment [19]. The arguments in a safety case are always accompanied by evidence supporting the arguments. More recently, there has been an OMG initiative called SACM aimed at standardising the notion of and the concepts related to assurance evidence and arguments [24]. While the above threads have been a useful start for the current SLR, they are either too specific (relating to only one standard or application domain) or do not provide a thorough and sufficiently detailed analysis of the possible evidence types and how to structure and assess them.

There are a number of SLRs in the literature whose scope partially overlaps with ours, e.g., on testing [2], on requirements specification [22], and on reliability [37]. None of these specifically address the topic of evidence for safety. Some past work attempts to compare safety standards in different domains with the aim of identifying the commonalities and differences among them [12,3,34]. However, these comparisons are limited in scope and, in contrast to ours, are not based on a systematic review.

In summary, little has been done to date by way of synthesising and summarising, in a comprehensive manner, the state of the art on safety evidence. Consequently, no unifying framework exists for reasoning about and communicating safety evidence. This observation led us to the need for the SLR as a way to gain new insights into how to specify, structure and assess safety evidence.

3. Research method

A SLR is a means of identifying, evaluating and interpreting available research relevant to a particular research question or topic area [18]. Individual studies contributing to a systematic review are called *primary studies*. A systematic review is a form of *secondary study*.

The purpose of a SLR is threefold [18]:

- To present a fair evaluation of a research topic by means of a rigorous and systematic methodology.
- To help in identifying any gaps in the current research in order to suggest further improvements.
- To summarise and provide background for new research activities.
- The design of the SLR reported in this paper started in October 2011. After several refinements and improvements, publication search was started in January 2012.

The following subsections present the research questions, the data sources, search strategies, the publication selection, and the quality criteria of the SLR.

3.1. Research questions

We formulated the following research questions (RQs)

RQ1. What information constitutes evidence of compliance with safety standards?

The aim of this question is to identify the various pieces of information such as artefacts, tool outcomes, and techniques considered as or used to provide evidence about the safety of a system during certification. The results are used to develop an evidence classification.

RQ2. What techniques are used for structuring evidence to show compliance with safety standards?

The aim of this question is to determine how the evidence collected during the various stages of a system's lifecycle can be structured and presented in a suitable way to demonstrate compliance with a safety standard.

Please cite this article in press as: S. Nair et al., An extended systematic literature review on provision of evidence for safety certification, Inform. Softw. Technol. (2014), http://dx.doi.org/10.1016/j.infsof.2014.03.001

RQ3. What techniques are used for assessing the adequacy of evidence?

The aim of this question is to determine how the evidence collected can be assessed for adequacy and for gaining confidence that it satisfies the safety requirements of a standard, and thereby confidence in the overall safety of a system.

RQ4. What challenges and needs have been the target of investigation in relation to safety evidence?

The aim of this question is to identify the various challenges addressed in the literature regarding the provision of evidence for safety certification. The results obtained will be useful to identify emerging trends and provide an overall view of the problems tackled in the literature.

RQ5. What commonalities exist among different application domains with regards to RQ1–RQ4?

The aim of this question is to identify, through a domain analysis, the similarities that exist among different application domains in terms of safety evidence provision. This research question is particularly relevant to practitioners who are engaged in cross-domain certification of components used across multiple application domains, or in assessing the feasibility of product reuse from domains other than that of the application they are working on.

3.2. Source selection

We performed two types of search to find publications relevant to the scope of the review. The first type was an *automatic search* performed on the following publishers' databases: ACM (portal.acm.org), IEEE (ieeexplore.ieee.org), Springer (springerlink.com), Elsevier (sciencedirect.com), and Wiley (onlinelibrary.wiley.com). We also used Google Scholar (scholar.google.com).

The second type was a *manual search* on the following workshops, conference, and journals: Australian Workshop on Safety Critical Systems and Software, High-Assurance Systems Engineering (HASE), IET System Safety, International Symposium On Leveraging Applications of Formal Methods, Verification and Validation (ISoLA), International Symposium on Software Reliability Engineering (ISSRE), International Conference on Computer Safety, Reliability and Security (SAFECOMP), Safety Critical System Symposium, Reliability Engineering & System Safety, IEEE Transactions on Reliability, and IEEE Transactions on Software Engineering. These venues correspond to conferences, workshops, and journals in which we repeatedly found, during our pilot automatic searches, publications that were relevant to the SLR. The decision about which venues to consider for manual search was made based on the authors' collective observations during the pilot searches, while we were elaborating the search strategy and before the search string was finalised. We did not consider satellite workshops at the conferences we manually searched.

In addition, expert knowledge was used for publication selection. We included relevant publications of which the authors were aware either on their own or because of having been informed by a colleague, but that had not been identified through the automated and manual searches. These were mainly studies that were accepted for publication but not yet available from the publishers when the automatic search was performed. In either case, publications added through expert knowledge were subject to passing the same inclusion criteria applied to automatic and manual searches.

3.3. Search string

We developed the search string by specifying the main terms of the phenomena under investigation. A number of pilot searches were performed to refine the keywords in the search string using trial and error. We removed terms whose inclusion did not yield additional papers in the automatic searches. After several iterations, we settled on the following search string. This search string, which is expressed as a conjunction of three parts, was used to search within keywords, title, abstract and full text of the publications[1]:

[part I]
("critical software" OR "critical system" OR "critical equipment" OR "critical application" OR "embedded system" OR "embedded software")
AND

[part II]
("safety certification" OR "safety evaluation" OR "safety assurance" OR "safety assessment" OR "safety qualification" OR "safety analysis" OR "safety standard" OR "safety requirement")
AND

[part III]
(evidence OR "safety case" OR "safety argument" OR "assurance case" OR "dependability case")

The first part of the search string captures keywords related to safety–critical systems. The second part concerns safety certification. Here, we consider several keywords in addition to "safety certification". These additional keywords capture terms that are sometimes used interchangeably with certification (e.g., safety evaluation), activities that share the same underlying principles as certification (e.g., qualification), and elements that serve as the main prerequisites to certification (safety standards and safety requirements). The third and final part of the search string relates to safety evidence. Here, we further consider an important context, namely safety cases and arguments, where safety evidence regularly appear without necessarily making a reference to the term "evidence". To this end and in line with what we observed in our pilot searches, we consider the fact that many papers have used the broader notions of assurance case and dependability case as synonyms for safety case, although these broader notions refer not only to safety but also to other dependability criteria such as security and reliability [16].

3.4. Study selection strategy and inclusion criteria

We specified inclusion and exclusion criteria for selecting primary studies. The basic inclusion criterion was to identify and select peer-reviewed studies related to safety assessment or certification of computer-based critical systems that dealt with safety evidence for showing compliance with safety standards. We searched and included publications written in English that provided information, artefacts, tool outcomes, or techniques considered as evidence for safety certification. When performing the manual search, we considered only those studies that had not been identified in the automatic search. In the journals, we only considered volumes from 1990 until the date when the automatic and manual searches were performed (January 2012). This was the publication year of the oldest paper found with automatic search and with manual search of conferences and workshops.

We also applied the following exclusion criteria, filtering out publications that matched any of the criteria:

[1] Where applicable, plural forms of the keywords were added to the queries performed over the publishers' databases. These plural forms are not shown in the search string to avoid clutter. In the case of SpringerLink and Google Scholar, where the search string was too long for the search engines, we performed the search through several sub-strings (12 sub-strings for SpringerLink and 21 sub-strings for Google Scholar).

Please cite this article in press as: S. Nair et al., An extended systematic literature review on provision of evidence for safety certification, Inform. Softw. Technol. (2014), http://dx.doi.org/10.1016/j.infsof.2014.03.001

- Grey literature, e.g., technical reports, working papers, project deliverables, and PhD theses
- Books, tutorials or poster publications
- Publications that addressed generic safety analysis techniques (e.g., FTA) but did not address provision of evidence for safety certification
- Papers in the context of non-computer based critical systems
- Publications whose text was not available

Study selection was performed through two main processes. The first process, reported in [21], covered all the sources (Section 3.2) except Google Scholar. In the second process, Google Scholar was considered as well as some new papers identified through expert knowledge.

The first process consisted of four phases. These phases are shown in Table 1 (represented as P1, P2, P3 and P4 in the table). In Phase 1, we applied the search string to the electronic databases, and a total of 2200 results were retrieved. In Phase 2, the first author read the abstract of the retrieved publications to determine their relevance to the scope of the SLR. The basic selection criterion at this stage was to check if the abstracts referred to safety evidence information for assessment or certification purposes or included the word evidence or some way to specify evidence (safety, assurance, or dependability case, or safety argument). During this phase, the first author also performed the manual searches on the selected conferences and journals. The same selection criteria as above were used for manual searches. From the 2200 studies obtained in the automatic search, 151 publications were selected. Performing the manual search resulted in the selection of 65 studies, making a total of 216 individual studies for the next phase.

In Phase 3, the studies were reviewed in depth. The workload was divided among the authors, with the first author being responsible for reviewing most of the studies. The remaining authors helped and provided guidance. No evidence information was initially found in 56 studies and these were excluded from the review.

In Phase 4, the second author performed two reliability checks. First, he randomly checked approximately 10% of the studies of Phase 1 by reading the abstract. Second, he inspected all the 56 papers excluded in Phase 3. At this stage, we regarded duplicates as those papers with at least one author in common that provide equivalent answers to the research questions (e.g., an extended version of a previous paper). In all cases, the extended and most recent version of the paper was included to extract maximum information. Excluded work considered to be potentially relevant was brought up for discussion and reviewed again. As shown in Table 1, eight studies were added as a result of the discussion and the relevant data was extracted from them. In addition, four studies were removed as a result of duplication. At this stage, seven papers were also added based on expert knowledge. These are studies that the authors considered to be relevant to the review and were not previously captured in any of the automatic or manual searches. The final number of primary studies at the end of this phase was 171.

To maximise the reliability of the SLR, we conducted a second publication selection process following the completion of the first publication selection process and the extraction of relevant data from the primary studies identified in the first process. In the second process, *Google Scholar* was used as the source for automatic search. This second process was meant as a confirmatory measure to increase confidence in the generalizability of the (earlier-obtained) findings from the first process. More specifically, the second process aimed to ensure that the key observations made based on the first process were not volatile, in the sense that the observations would no longer be valid in light of new findings.

The second publication selection process consisted of four steps, shown in Table 2 (represented as S1, S2, S3 and S4). In step 1, when we applied the search string, we obtained a total of 5430 studies.[2] Since the inclusion of Google Scholar was to further mitigate the risk of having missed relevant publications and information, we only checked over half the studies (2763). In step 2, we excluded publications that were from any of the publishers' sites previously checked and also those matching the exclusion criteria (grey literature, technical reports, etc.). This resulted in the selection of 97 studies. In step 3, the second author selected 49 studies after reading the abstract. These studies, which had not been identified through the first selection process, were all peer-reviewed publications listed on webpages of universities, organisations, research associations, or small publishers. In step 4, the first author performed a full text review of these 49 studies and selected 39 as primary studies. Additionally, 7 papers were added based on expert knowledge during this second publication selection process.

The two publication selection processes outlined above collectively resulted in 171 + 47 = 218 primary studies for the SLR.

3.5. Data extraction and quality criteria

We designed a data extraction template (a spreadsheet) to collect the information needed to answer the research questions. Apart from the *bibliographic information* (title, authors, year, and publisher), we extracted from each study the *application domain* in which the system under assessment or certification was used, the *underlying safety standard(s)* used to show compliance, the *information, artefact, tool, or technique contributing to evidence*, *techniques for evidence structuring*, *techniques for assessing confidence* on the evidence collected, and the *needs and challenges addressed* about provision of evidence. Appendix A provides a table with some sample data extracted from the studies. All the information about the data extracted from all the studies can be found in [20].

We further extracted data for publication quality assessment. For this, we defined three criteria:

- **Evidence abstraction level**, which was assigned on the basis of the specificity of the evidence instances presented in a given study. The levels allow us to weight the quality of evidence items identified from the analysis of the primary studies. The abstraction levels defined, from the most abstract to the most specific, were: *generic, domain level, safety standard level, system type level*, and *specific system level*. Using the evidence types from our evidence classification (Section 4.1), example instances of evidence for the non-generic abstraction levels are: *Hazard specification* for domain level (e.g., nuclear domain) [PS98], *Source code* for safety standard level (e.g., for DO-178B [PS172]), *System Historical Service Data Specification* for system type level (e.g., COTS-based systems [PS170]), and *Model Checking Results* for specific system level, e.g., instantiated for a specific pacemaker software [PS84]. The "generic" abstraction level refers to instances of evidence mentioned in a primary study that are not presented within the scope of any specific domain, standard, system type, or specific system. Generally, we consider lower abstraction levels and thus more specific evidence to be more useful since it is more likely for those studies to contain some practical advice.
- **Validation method**, which was assigned based on how a given study had been validated. The studies were classified as: *case study* (validated during projects by practitioners different from

[2] Performing an automatic search for publications in Google Scholar had two main constraints. First, Google Scholar allows access (to read the content) only for the first 1000 results of a search. Second, the search engine permits only a limited length search string. In order to obtain only 1000 results per search and have a search string of acceptable length, we used a number of separate sub-strings that were based on the original search string. The sub-strings were a result of different combinations of the three parts of the main string (Section 3.3).

Table 1
SLR phases and number of publications in conference version.

Source	P1: Studies investigated	P2: Studies selected after reading abstract	P3: Studies selected after reading full text	P4: Studies finally selected
IEEE (Publisher)	775	75	60	67
ACM (Publisher)	125	15	11	10
Elsevier (Publisher)	448	22	14	14
Springer (Publisher)	689	33	21	22
Wiley (Publisher)	163	6	4	4
Australian Workshop on Safety Critical Systems and Software	–	7	4	4
HASE (Conference)	––	0	0	0
IET System Safety (Conference)	–	12	8	8
ISoLA (Conference)	–	4	3	3
ISSRE (Conference)	–	2	2	2
SAFECOMP (Conference)	–	20	17	14
Safety Critical System Symposium (Conference)	–	14	12	12
Reliability Engineering & System Safety (Journal)	–	4	3	3
IEEE Transactions on Reliability (Journal)	–	0	0	0
IEEE Transactions on Software Engineering (Journal)	–	2	1	1
Expert Knowledge	–	–	–	7
	2200	216	160	171

Table 2
Publication selection process and number of publications in goole scholar.

Source	S1: Studies investigated	S2: Studies selected after applying exclusion criteria	S3: Studies selected after reading abstract	S4: Studies finally selected
Google scholar	2763	97	49	39
Expert knowledge	–	–	–	8
	2763	97	49	47

the authors), *field study* (validated with data from real projects, but not during the execution of the projects), *action research* (validated during real projects by the authors themselves), *survey* (validated on the basis of practitioners' opinion and perspectives), or none. It is important to note that we use the term "validation" in a broad sense. In particular, validation does not necessarily imply validation in a controlled environment such as a controlled experiment. Indeed, we did not find any primary studies reporting a controlled experiment. Nonetheless, we consider information gathered from validated work to be more useful as they better reflect the state of practice.

– **Tool support,** which assists in the provision of evidence (collection, structuring, and assessment) for certification or safety assurance purposes. We consider the availability of tool support to be an important maturity factor for the underlying technique and a necessary step for its industrial application.

4. Results

This section presents the results of the review, answering the research questions individually based on the extracted data from the 218 studies over a publication period of 22 years. With respect to the application domains and the safety standards referred to in the studies, we identified eight application domains and 16 safety standards.

Fig. 1 shows: (a) the number of primary studies published from 1990 to 2011; (b) the number of papers found for each domain, and (c) the number of papers per safety standard referred to in the literature. Publications during the year 2012 are not shown in the Fig. 1(a) since this was the year the search was performed and would represent partial numbers. The eight application domains identified in the studies are:

(1) *Aerospace* dealing with systems in crafts that fly in the atmosphere and outer space.
(2) *Aviation* dealing with aircrafts systems that fly in the troposphere.
(3) *Automotive* dealing with systems that run on motor-vehicles on the road.
(4) *Maritime & (Offshore) Energy* dealing with systems in ships and offshore units, and for oil, gas, and offshore natural resource extraction.
(5) *Medical* dealing with systems in medicine and healthcare.
(6) *Nuclear* dealing with systems in nuclear power plants and controllers.
(7) *Railway* dealing with rail-road systems that run on tracks.
(8) *Robotics* dealing with the design, construction, operation, and application of robots.

Note that in Fig. 1, we do not include studies that mention more than one domain or safety standard. Although some of the domains or standards in these studies are within the scope of the SLR, we could not conclusively determine the domain or standard to which the relevant information (evidence information, technique, tools) would correspond.

4.1. What information contstitues evidence of compliance with safety standards?

We created a taxonomy, shown in Fig. 2, for evidence types based on the various evidence examples, artefacts, tools and techniques found in the primary studies. A taxonomy provides an intuitive and yet comprehensive way to present and summarise the fraction of the results having to do with evidence information requirements, especially considering the vast amount of informa-

Please cite this article in press as: S. Nair et al., An extended systematic literature review on provision of evidence for safety certification, Inform. Softw. Technol. (2014), http://dx.doi.org/10.1016/j.infsof.2014.03.001

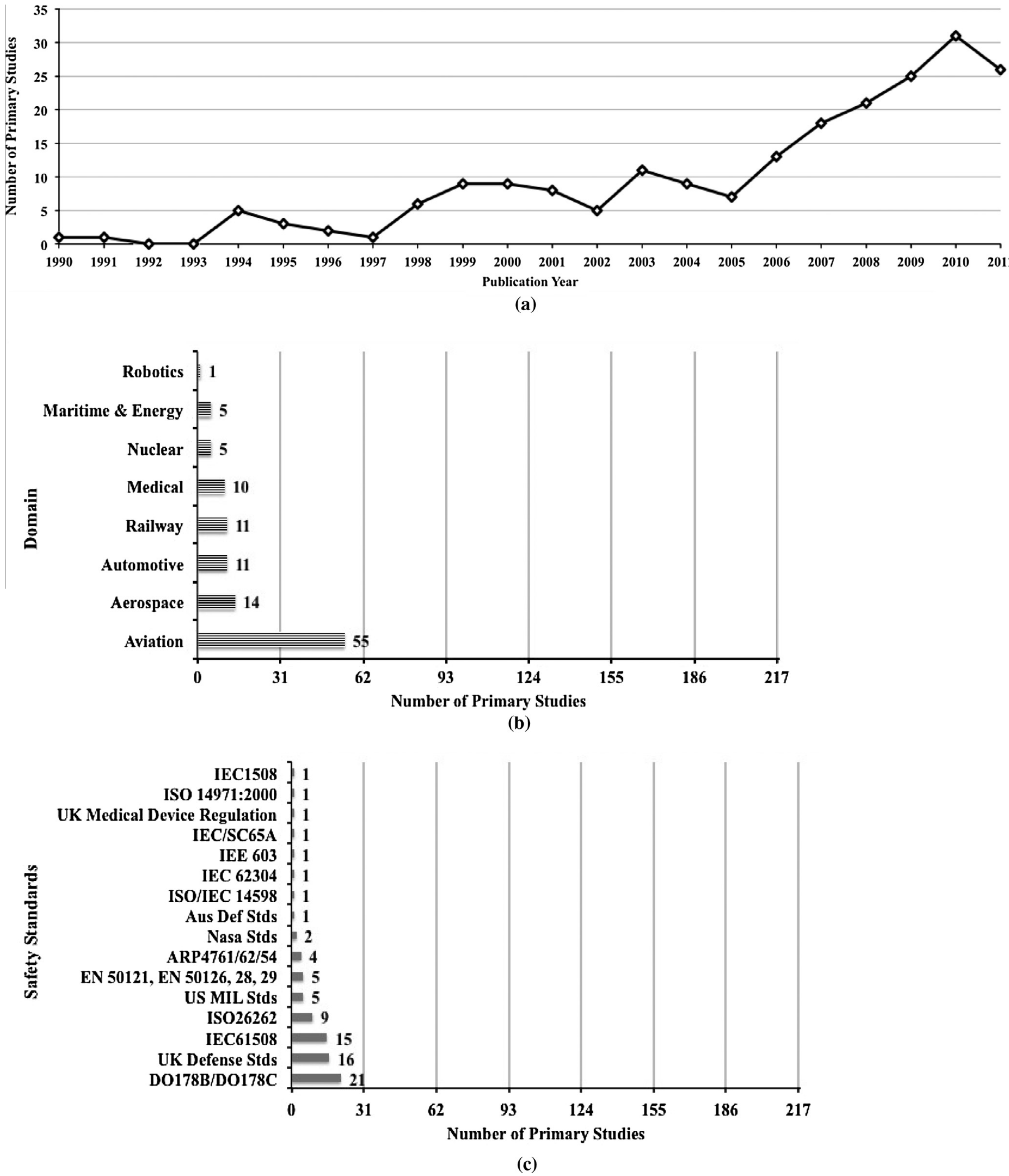

Fig. 1. (a) Number of studies per publication year, (b) number of studies per application domain, (c) number of studies per safety standard.

tion found in the primary studies (see Appendix B). Moreover, the taxonomy is an effective means for communicating the results in a more structured manner. Several iterations were made before the current structure of the taxonomy was developed. Experts in system safety and certification reviewed and provided feedback on the extracted evidence types.

The taxonomy contains 49 different basic evidence types, denoted as leaf nodes in Fig. 2. Each leaf node in the taxonomy has been referred to by at least two primary studies. The taxonomy is complemented by a glossary given in Appendix B. The glossary provides some clarifications to ensure a better understanding of the taxonomy and how it was built. The glossary also provides

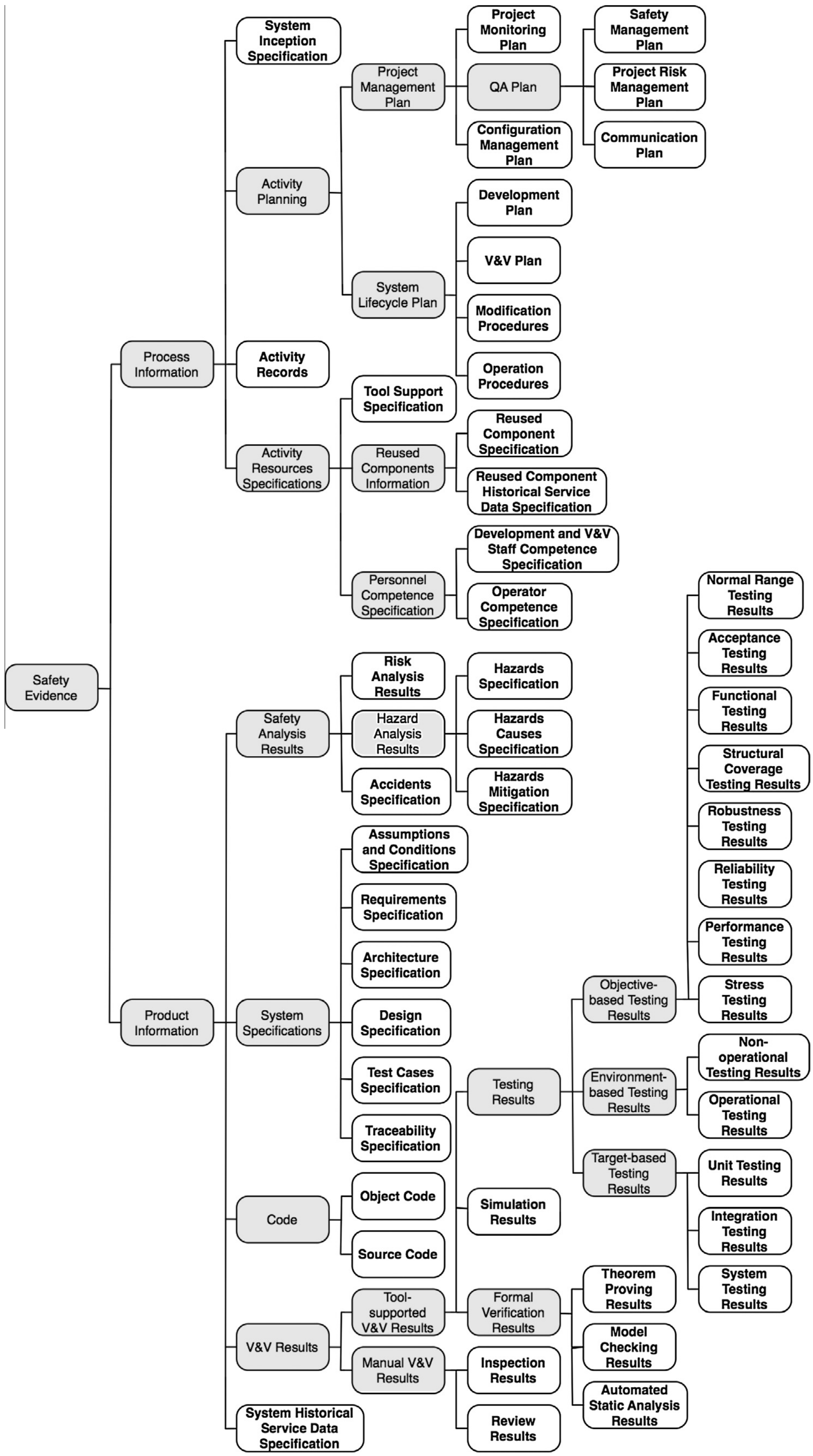

Fig. 2. Evidence taxonomy.

(1) a definition for each basic evidence type, (2) the source(s) on which the definition is based (different from safety standards), (3) the synonyms identified in the literature for each evidence type, and (4) the tools, techniques, artefacts, and information considered as or used to provide evidence in the literature. The full list of extracted data from each primary study and citations are available in [20].

Table 3 provides the information regarding the number and percentage of studies in which each evidence type was identified. Since different studies had information at different abstraction levels (Section 3.5), we denote the lowest abstraction level identified for each evidence type in the table.

Our results indicate that the most frequent evidence types referred to in the literature are *Hazards Cause Specification* (appearing in 111 out of 218 papers, i.e., 51%), *Risk Analysis Results* (51%), *Hazard Specification* (43%), *Accident Specification* (34%), *Requirements Specification* (24%), *Hazards Mitigation Specification* (23%), and *Design Specification* (20%). The least frequent types are *Communication Plan* (1%), *System Testing Results* (1%), *Object Code* (1%), *Non-operational Testing Results* (1%), *Project Risk Management Plan* (2%) and *Normal Range Testing Results* (2%). Only *Communication Plan* has not been mentioned in studies that have been validated. The above frequencies indicate that the evidence types under *Safety Analysis Results* (in Fig. 2) are the most common.

4.2. RQ2: What techniques are used for structuring evidence to show compliance with safety standards?

In 117 of the 218 selected studies, we identified some technique for structuring safety evidence. We divide the techniques into three main categories, described below. The percentage given for each category is the rate of papers in that category over the 107 relevant papers. Some studies referred to more than one technique.

1. *Argumentation-Induced Evidence Structure* (92%): Argumentation is an approach that communicates the reasons why a system is considered to be acceptably safe. The structure of the argumentation induces a specific way to structure the evidence, as arguments need to be supported by evidence that *directly* substantiates them. The structure induced as the result of the

Table 3
Evidence type identified in the primary studies.

Evidence type	Number of papers	Percentage of papers (%)	Lowest abstraction level
Hazard causes specification	111	51	Specific system level
Risk analysis results	111	51	Specific system level
Hazard specification	93	43	Specific system level
Accidents specification	75	34	Specific system level
Requirements specification	52	24	Specific system level
Hazards mitigation specification	51	23	Specific system level
Design specification	43	20	System type level
Review results	37	17	Specific system level
Structural coverage testing results	36	17	Specific system level
System historical service data specification	27	12	Specific system level
Traceability specification	27	12	Specific system level
Development and V&V staff competence specification	26	12	Specific system level
Reused component historical service data specification	26	12	Specific system level
Simulation results	25	11	Specific system level
Model checking results	24	11	Specific system level
Unit testing results	24	11	Safety standard level
Automated static analysis results	23	11	Specific system level
Architecture specification	22	10	Specific system level
Development plan	22	10	Specific system level
Integration testing results	20	9	Safety standard level
Reliability testing results	20	9	Specific system level
Activity records	18	8	Specific system level
Functional testing results	18	8	Safety standard level
Modification procedures plan	17	8	Specific system level
V&V plan	16	7	Specific system level
Inspection results	15	7	Specific system level
Operation procedure plan	15	7	Specific system level
Safety management plan	15	7	Specific system level
Source code	15	7	System type level
Configuration management plan	14	6	System type level
Performance testing results	14	6	Specific system level
Theorem proving results	14	6	Specific system level
Reused component specification	13	6	Specific system level
Robustness testing results	13	6	Specific system level
Stress testing results	12	6	System type level
Operator competence specification	11	5	Specific system level
Tool support specification	11	5	Safety standard level
Operational testing results	10	5	Specific system level
Acceptance testing results	9	4	Specific system level
Assumptions and conditions specification	8	4	Specific system level
System inception specification	7	3	Specific system level
Project monitoring plan	6	3	System type level
Test cases specification	6	3	Specific system level
Normal range testing results	5	2	Specific system level
Project risk management plan	5	2	Safety standard level
Non-operational testing results	3	1	Specific system level
Object code	3	1	Safety standard level
System testing results	3	1	Safety standard level
Communication plan	2	1	Domain level

Please cite this article in press as: S. Nair et al., An extended systematic literature review on provision of evidence for safety certification, Inform. Softw. Technol. (2014), http://dx.doi.org/10.1016/j.infsof.2014.03.001

argumentation can be expressed either *graphically* or *textually*. In the graphical sub-category, we identified the following techniques:

– GSN (e.g., [PS3, PS5, PS8, PS9, PS10]), which can be used to document explicitly the elements and structure of an argument and the argument's relationship to evidence. In GSN, the claims of the argument are documented as goals and items of evidence are documented in solutions.
– CAE (e.g., [PS20, PS22, PS72, PS78]), which promotes a three-tiered approach similar to GSN, composed of a top-level claim asserted within an argument, a description of the arguments presented to support a claim, and a reference to the evidence that is presented to support a claim or argument.
– BBN (e.g., [PS23, PS38, PS58, PS175, PS178]), which induces a structures to evidence in a directed acyclic graph representing the conditional dependencies among them.
– KAOS, which is a goal modelling language that has also been used for safety case specification [PS137, PS208]. This approach decomposes top-level goals using AND/OR operators in an argumentation-like way until evidence of goal achievement is provided.
– SSG [PS138], which are linear graphs that represent a safety specification as nodes and evidence and relationships among them as edges.

In the textual sub-category, we include studies that use a structured text-based presentation of the arguments and the evidence supporting them. We identified the following techniques in the textual sub-category:

– Trust Cases [PS176,40], which induce a structured textual format for safety claims, arguments, and evidence presenting them as assumptions with references to documents.
– Structured HTML [PS185], which uses HTML tags to link and structure the various artefacts used as evidence for safety.
– Structured text [PS80], which proposes several possible approaches namely: structured prose, which introduces a certain structure to a normal prose by requiring that the critical parts of the argument be explicitly denoted; argument outline, which uses indentation, numbering, and different fonts to structure arguments and evidence adopting an outline format; mathematical proof, which uses the geometric proofs structure (given, statements, and reasons) used in mathematics; and, LISP style, which uses the syntax structure of the LISP programming language with short names and parentheses for evidence and arguments.

2. *Model-Based Evidence Specification (5%):* We classify in this category those techniques that characterise the structure of safety evidence using models. We identified the following approaches in the studies:
   – Sector-specific UML meta-models [PS54,PS122] and UML profiles built specifically for standards such as DO-178B [PS172] and IEC61508 [PS121].
   – Data modelling using entity-relationship diagrams to structure the data content in large safety cases including the evidence aspects [PS99].
   – Process models capturing the activities or processes that produce the artefacts used as evidence and present them using a tree-based structure [PS67].
3. *Textual Templates (3%):* These templates provide predefined sections or tables along with constraints for structuring evidence in a predefined textual format. We identified the following approaches:
   – The CENELEC template [PS51,PS118], which is used in the railway domain for structuring evidence in a series of reports such as quality management reports and safety management reports.
   – The ACRuDA template [PS50], which is used to structure evidence according to a pre-defined safety case structure.
   – Template Add-ons [PS19], which provides a template for predefined set of documents that are to be produced at different system development and safety assurance phases. It also provides suggestions on the required approaches for documentation, semi-formal description, and verification and validation procedures.

Fig. 3 shows the number of studies that refer to each evidence structuring technique. Two clarifications need to be made in relation to the evidence structuring techniques identified. First, we did not consider *unstructured text* because it does not provide means for systematically organising evidence information. Second, in the Model-Based Specification category, we only considered techniques that are aimed at specifying the structure of the evidence, as opposed to the structure of, for instance, the system that the evidence is generated or used for. For example, AADL [PS56] has been used for modelling the architecture and design of safety–critical systems, but not for modelling the structure of safety evidence. Hence, AADL was not considered as an evidence structuring technique. In contrast, UML, due to its broader expressiveness, has been used for modelling both systems and safety evidence, and was hence considered.

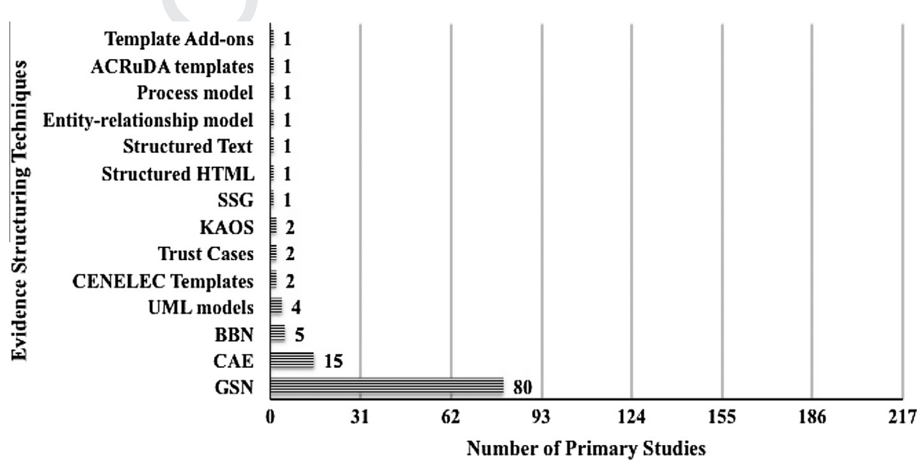

Fig. 3. Number of studies referring to each evidence structuring technique.

4.3. RQ3: What techniques are used for assessing the adequacy of evidence?

We identified techniques for evidence assessment in 105 of the total 218 studies. We classify these techniques into four categories. The percentage given for each category is the rate of papers in that category over the 105 relevant papers. Some studies referred to more than one type of technique.

1. *Qualitative Assessment (68%):* We classify techniques that use non-numerical methods for assessment of evidence in this category. Argumentation (e.g., [PS1,PS7,PS11,PS14,PS30]) is the most widely identified technique under qualitative assessment. Argumentation can be based on unrestricted natural language, (semi-) structured natural language, or graphical argumentation structures such as GSN. Graphical argumentation structures generally have the advantage of being easier to understand, review, and navigate. Argumentation can be enhanced by "qualitative tags" that capture the level of trustworthiness of evidence. The approaches that we found for this purpose are:
   – Safety Evidence Assurance Levels (SEAL) [PS57], providing four levels to capture the degree of confidence in safety evidence, the highest level of assurance being incontrovertible, followed by compelling, persuasive, and the lowest level being supportive.
   – Safety Assurance Levels (SAL) [PS128,PS162,PS170], which are similar to SEALs but also address confidence propagation rules between arguments and sub-arguments.

   –Our review also identified qualitative methods for assessment that are not based on argumentation. These are:
   – Activity-based quality model [PS83], which uses quality matrices to assess evidence for compliance with the IEC62304 standard.
   – Evidence-confidence conversion process [PS171], which assesses safety evidence through a review process that results in the specification of the confidence in the safety of the system.

2. *Checklists (16%):* We classify in this category techniques that introduce a "*to-do list*" consisting of a set of guided questions that need to be answered or checked while reviewing the evidence. The questions could, for example, be a set of conditions that must be met in order to gain confidence in the evidence collected and to check its sufficiency [PS66]. We identified different variations of checklists in the literature:
   – Design Checklists [PS114], which assess evidence based on the design of the system.
   – GQM-based checklists [PS47], which are based on the Goal/Question/Metric measurement framework [6]. They define top-level goals for assessing product and process evidence, questions to be answered to achieve these goal and metrics providing a measurable reference against which analysis can be performed.
   – Argumentation-based checklists [PS109], which assess evidence by mixing checklists with argumentation.
   – The Taxonomy-Based Questionnaire [PS79], which contains 305 questions addressing the safety attributes and artefacts in the Software Safety Risk Taxonomy and Software Safety Risk Evaluation process [14].
   – Plain Checklists [PS50], which are checklists that do not fall under any of the more specific variations discussed above.

3. *Quantitative Assessment (10%):* We classify in this category techniques that use numerical measures for assessment of evidence. These techniques are:
   – BBNs (e.g., [PS41,PS101,PS134,PS167,PS168]), which assess evidence in the presence of uncertainty by using conditional probability distributions. This technique is used in conjunction with BBN structuring of evidence (Section 4.2). This is the most frequent quantitative technique in the literature for evidence assessment.
   – The Modus approach [PS137], which combines quantitative assessment with formal argumentation structures. The approach is based on quantitative reasoning that uses goal models (KAOS), expert elicitation, and probabilistic simulation for assessing the overall goal of a safety case.
   – Evidence Volume Approach [PS96], which allows an internal expert to assign weighted factors on evidence that describe the relative importance of each piece of evidence. An aggregate function is then chosen for the weighted evidence to calculate a volume known as evidence volume, based on which an outcome (accept or reject) is chosen.

3. *Logic-based Assessment (6%):* In this category, we classify techniques that use logical formulae, such as first-order logic statements, to articulate and verify the properties of interest over evidence items and their relationships. Logic-based techniques are best suited for checking the well-formedness and consistency constraints of evidence information. For example, OCL [23] has been used to ensure that there is a consistent link between the evidence items produced for a particular system, and that the evidence items required by a safety standard are available [PS122,PS82,PS83,PS121,PS122,PS131].

Fig. 4 shows the number of studies that refer to each evidence assessment technique. It is important to make the following clarifications about the evidence structuring techniques identified. First, in the literature, *expert judgment* can and has been used in conjunction with all the techniques outlined above. However, we have not regarded expert judgment per se as an assessment technique. For expert judgment to have any credibility, the rationale behind it must always be made explicit (e.g., through assumptions or argumentation). Second, we do not regard assignment of integrity levels such as SIL as a technique for evidence assessment. These levels are concerned with the assessment of the integrity of the product that the evidence relates to, not the integrity of evidence itself.

4.4. RQ4: What challenges and needs have been the target of investigation in relation to safety evidence?

We identified several categories of general challenges and needs related to providing safety evidence information and to structuring and assessing the evidence. Some primary studies note more than one need or challenge. Although not all the corresponding primary studies are referenced in each category, examples are provided to better understand how the primary studies were categorised. The categories of challenges and needs addressed in the literature are as follows:

(1) *Specification of evidence content:* The challenge that was noted the most (60 papers out of 218) was determining in a systematic way *what* information was necessary to be provided as evidence in a given domain and for compliance with a particular set of applicable standards. For example, Habli & Kelly [PS69] address the challenge of finding the right balance between product-based and process-based evidence for certification. Similarly, Bate et al. [PS12] investigate the challenge of identifying supporting evidence when modern super-scalar processors are used

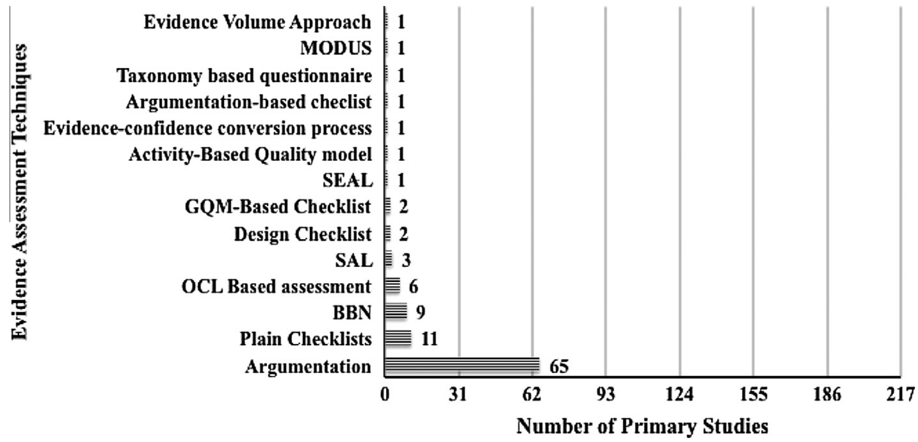

Fig. 4. Number of studies referring to each evidence assessment technique.

in the current safety–critical systems. We think that the evidence taxonomy built as a response to RQ1 can help tackle this challenge.

(2) *Construction of safety cases:* The second most identified challenge (57 papers) relates to the development of safety cases, particularly providing methodological guidance for safety case construction and decomposition of the arguments and the evidence in a way that permits more precise and cost-effective demonstration of compliance. The need for well-defined structures for claims, arguments, and evidence relates to the structuring techniques identified in RQ2. For example, Bishop et al. [PS20] acknowledge the importance of constructing well-defined safety cases to minimise safety and commercial risk. They propose a top-down approach for safety case development that structures safety cases in layers to accommodate changes in them. Similarly, Feather & Markosian [PS55] discuss the challenge of building safety cases for NASA's safety–critical space software and provide guidance to help future developers of safety cases for similar software systems.

(3) *Capturing the degree of credibility or relevance of the evidence:* We identified 31 papers in which researchers acknowledged that different evidence items could have different levels of credibility depending on their source, or different degrees of contribution towards the satisfaction of different compliance requirements. To capture credibility or relevance, one might need to be able to assign weights to the evidence items or to the links between evidence items and safety arguments or claims. For example, Bouissou et al. [PS23] use BBN for helping assessors weight the evidence provided by using probability distribution functions. Similarly, Czerny et al. [PS37] discuss the challenge of providing convincing evidence of safety for "by-wire systems" in the automotive domain. This represents a major technology change demanding higher levels of analysis, design, and verification. Techniques identified in RQ3 for evidence assessment relate to this need.

(4) *Better development processes and better evidence about process compliance:* Among the selected primary studies, 30 noted the need for better development processes of safety–critical systems, thereby making it easier to rigorously verify that the development process followed is in compliance with the applicable safety standards. For example, Habli & Kelly [PS67] use a model-based approach to define an extendable metamodel for describing the lifecycle process and reliability assurance process by enabling automatic verification of compliance with safety standards. In another example, Hall and Rapanotti [PS71] introduce the concept of assurance-driven design for system development, which regards assurance arguments or assurance cases as important as the product itself.

(5) *Ambiguities of safety standards:* We identified 25 primary studies citing ambiguities (or problems) in the application of standards, such as the existence of multiple interpretations of the evidence requirements in the standards. These studies also provided guidance on how to show compliance with a single standard or a set of standards. For example, Evans et al. [PS53] explore the evidence requirements and its sufficiency for the UK defence standard 00–56, and compare them with civil standards such as DO-178B, ARP4754, ARP4761, and IEC 61508. Dittel & Aryus [PS46] discuss the challenges of interpretation, implementation, and identification of the right level of detail when building safety cases for compliance with ISO 26262.

(6) *Certification of systems made up of components and subsystems:* We identified 17 papers that mentioned challenges related to the construction, structuring, and assessment of evidence for systems that reuse existing components or subsystems such as legacy or Commercial-Off-The-Shelf (COTS) software. For example, Fan and Kelly [PS170] propose a contract-based approach for justifying the use of COTS in safety–critical systems. The approach evaluates application-specific safety requirements against corresponding assurance requirements derived from the COTS. Esposito et al. [PS52] propose another systematic approach for qualification and selection of COTS based on a customised quality model that can guide and evaluate COTS selection.

(7) *Need for providing argumentation:* We identified nine papers that address the importance of demonstrating and justifying how evidence collected supports safety claims through argumentation. For example, Linling & Kelly [PS100] explore the need for a clear and defensible arguments and potential issues of argumentation-based assurance in aircraft certification. Clegg [PS32] discusses how faults and failures can be introduced into a FPGA, what possible mitigation techniques can be used, and the need for arguments to demonstrate how a FPGA meets its safety requirements.

(8) *Demonstration of compliance for novel technologies:* Seven papers cited challenges related to provision of evidence for certification of systems that make use of technologies that are novel for safety–critical domains. For example, Daniel & Mario [PS139] discuss how new computing trends

Please cite this article in press as: S. Nair et al., An extended systematic literature review on provision of evidence for safety certification, Inform. Softw. Technol. (2014), http://dx.doi.org/10.1016/j.infsof.2014.03.001

Table 4
Evidence types in different application domains.

Evidence types	Aerospace	Automotive	Aviation	Medical	Maritime	Nuclear	Railway	Robotics	Rate (%)
Acceptance testing results	–	–	X	–	–	X	X	–	38
Accidents specification	X	X	X	–	X	X	X	X	88
Activity records	X	X	X	X	X	X	X	–	88
Architecture specification	X	X	–	–	–	X	–	–	38
Assumptions and conditions specification	–	X	X	X	–	–	X	–	50
Automated static analysis results	X	X	X	X	–	X	X	X	88
Communication plan	–	–	X	X	–	–	–	–	25
Configuration management plan	–	X	X	–	–	–	X	–	38
Design specification	–	X	X	X	X	X	X	X	88
Development plan	X	X	X	–	X	X	X	–	75
Development and V&V staff competence specification	X	X	X	X	X	X	X	X	100
Functional testing results	–	–	X	X	–	–	–	–	25
Hazards causes specification	X	X	X	X	X	X	X	X	100
Hazards mitigation specification	X	X	X	X	X	X	X	X	100
Hazards specification	X	X	X	X	X	X	X	X	100
Inspection results	X	–	X	X	X	–	–	X	63
Integration testing results	X	X	X	–	–	X	X	–	63
Model checking results	X	X	X	X	–	X	–	X	75
Modification procedures plan	X	X	X	X	–	–	X	–	63
Non-operational testing results	–	–	X	X	–	–	X	–	38
Normal range testing results	–	X	–	–	–	–	X	X	38
Object code	–	–	X	–	–	–	–	–	13
Operation procedures plan	–	–	X	X	X	–	X	–	50
Operational testing results	X	–	X	–	–	–	–	X	38
Operator competence specification	–	–	X	X	X	–	–	X	50
Performance testing results	X	–	X	–	–	–	–	–	25
Project monitoring plan	–	–	X	X	–	–	–	–	25
Reliability testing results	X	–	X	–	–	X	X	–	50
Requirements specification	X	X	X	X	X	X	X	X	100
Reused component specification	–	–	X	X	–	X	X	–	50
Reused component historical service data specification	X	–	X	–	X	–	–	–	38
Review results	X	X	X	X	X	X	X	X	100
Risk analysis results	X	X	X	X	X	X	X	X	100
Project risk management plan	–	–	–	X	–	–	–	–	13
Robustness testing results	–	X	X	X	–	X	–	–	50
Safety management plan	–	–	–	–	X	–	X	–	25
Source code	–	–	X	X	–	X	X	–	50
Simulation results	X	X	X	–	X	X	–	X	75
Stress testing results	X	–	X	–	–	–	–	–	25
Structural coverage testing results	X	X	X	–	–	X	X	X	75
System historical service data specification	–	X	X	X	X	X	–	–	63
System inception specification	–	X	X	–	–	–	–	–	25
System testing results	X	–	X	–	–	–	–	–	25
Test cases specification	–	X	X	–	–	–	–	–	25
Theorem proving results	X	X	X	–	X	–	X	X	75
Tool support specification	–	–	X	–	–	–	–	–	13
Traceability specification	X	X	X	X	X	X	X	X	100
Unit testing results	X	X	X	X	X	X	X	X	100
V&V plan	–	X	–	X	–	X	–	–	38
Total	55%	59%	90%	57%	43%	53%	57%	41%	

like ubiquitous computing needs to be adaptive to react appropriately to dynamic changes to environment and user requirements. They also present details of conditional safety certificates to evaluate safety of adaptive systems. In a similar vein as the above, Rushby [PS136] discusses how novel technologies like adaptive systems modify and synthesise functions at runtime, and proposes a framework that uses runtime verification, thereby allowing certification to be partially performed at runtime.

(9) *First-time certification or recertification of "proven-in-use" systems:* We identified seven papers highlighting the challenge of certifying systems that have not been previously certified, or recertification of systems that previously invoked the "proven-in-use" principle but can no longer do so, e.g., due to tighter regulations or the fact that the systems evolved since they were last certified as proven-in-use. Proven-in-use here refers to the situation where there is convincing evidence, based on the previous operation of the system, that it meets the relevant safety requirements of a standard. For example, Cameron et al. [PS187] provide an approach for certification of UAS by demonstrating compliance to relevant proven-in-use UAS airworthiness codes. In another example, Meacham et al. [PS111] address the issue of applying traditional software safety standards to legacy safety–critical systems, with the aim of re-certifying the legacy systems. The paper proposes a model that captures relationships between pre- and post-modification software, and a framework that provides guidance on how to achieve airworthiness certification for the modified legacy software.

4.5. RQ5: What commonalities exist among different application domains with regards to RQ1–RQ4?

In this section, we compare the results obtained for RQ1–4 with the eight domains identified in the literature. We analyse which evidence types, structuring techniques, assessment techniques, and challenges have been addressed in each domain.

The rate information in the tables that follow (e.g., the last column of Table 5) specifies the percentage of domains in which a particular evidence type, technique, or challenge was found. The

Table 5
Evidence structuring techniques in different application domains.

Domain	Argumentation-induced evidence structure	Model-based evidence specification	Textual templates	Total (%)
Aerospace	X	X	–	67
Automotive	X	–	–	33
Aviation	X	X	–	67
Medical	X	–	–	33
Maritime & energy	X	X	–	67
Nuclear	X	–	–	33
Railway	X	–	X	67
Robotics	X	–	–	33
Rate	100%	38%	13%	

total (e.g., the last row of Table 5) specifies the percentage of evidence types, techniques, or challenges that have been found in a particular domain.

The x symbol shows that the particular evidence type, technique, or challenge has been found for a domain in at least one study. We did not consider for this analysis those studies that (1) indicate more than one domain or (2) do not explicitly specify the application domain that they target.

Table 4 provides the comparison for the evidence types. Nine types have been identified in all the domains: Development and V&V Staff Competence Specification, Hazards Causes Specification, Hazards Mitigation Specification, Hazards Specification, Requirements Specification, Risk Analysis Results, Review Results, Traceability Specification, and Unit Testing Results.

Table 5 presents a matrix of the categories of evidence structuring techniques and the application domains. *Argumentation-induced evidence structure* has been identified in all the domains. More than one structuring technique was identified in aerospace, aviation, maritime & energy, and railway domains.

Table 6 presents a matrix of the categories of evidence assessment techniques and the domains. *Qualitative assessment* has been identified in all the domains. Aviation includes all the four categories of evidence assessment techniques. Except Robotics, all domains have referred to at least two evidence assessment categories. The reason could be because we identified only one study in this domain.

Table 7 presents the matrix of identified challenges or needs in each of the application domains. Difficulties with categorising evidence information or specifying what evidence information is made of, and challenges with safety case construction have been reported in all the domains. Aviation has acknowledged all the eight categories of challenges.

4.6. Quality assessment

As discussed in Section 3.5, we defined three quality criteria for the selected primary studies. This section provides our findings in relation to these criteria.

With regards to evidence abstraction levels, we consider only the lowest (i.e., the most specific) level found in any given primary study. As shown in Fig. 5(a), the most frequent evidence abstraction level is "generic" (35%). Nevertheless, the remaining levels – which go beyond just providing generic examples – still collectively account for a majority of the publications (65%). This said, the lowest-level (and in our view the most useful) abstraction levels, namely system-type level and system-specific level, account only for 14% of the studies.

Fig. 5(b), shows the statistics for the validation methods used by the studies. The vast majority of studies (72%) have not been validated with practitioners, or with data from a real project. A small fraction of the studies (15%) have been validated in actual projects, by means of action research or case studies. The least used validation method is survey (2%).

The *Communication Plan* evidence type, three types of techniques from the *Argumentation-Induced Evidence Structure* (*Structured HTML*, *Structured Text*, and *Safety Specification Graphs*), and six evidence assessment techniques (*SEAL*, *SAL*, *Activity-based Quality Models*, *Evidence-Confidence Conversion Process*, *Taxonomy-based Questionnaire* and *Evidence Volume approach*) have not been mentioned in the studies that have been validated with the methods considered. All the challenges and needs identified in the literature have been noted in at least two studies that have been validated. More details are shown in Table 8. Please note that, as explained previously in Section 3.5, the term "validation" does not imply validation in a controlled experiment (e.g. controlled experiment).

With respect to tool support, 53 studies noted some tool for creating evidence information, structuring of evidence, or assessment of evidence. A total of 39 different tools were identified from these studies. Table 9 provides the list of tools and the number of studies in which each tool was validated. Only five tools were noted twice or more than twice in the validated studies.

5. Discussion

In this section, we discuss the implications of the results obtained from the SLR in the context of future research and of practice.

The results from the review provide a general research-oriented view on evidence provision. The evidence taxonomy built as part of the review depicts a holistic view of the development and verification artefacts and the information that constitutes safety evidence. We believe that this taxonomy is a useful reference to new researchers, helping them get better acquainted with the area.

The taxonomy captures, at an abstract level, the types of information that a safety evidence management tool should be capable of handling. One can use the taxonomy to elicit detailed requirements about the contents of each evidence type as well as the relationships that must be maintained between instances of different evidence types in a tool. Using these requirements, one can further elaborate the analysis scenarios for which tool support is required,

Table 6
Evidence assessment techniques in different application domains.

Domain	Qualitative assessment	Checklists	Quantitative assessment	Logic-based assessment	Total (%)
Aerospace	X	X	–	–	50
Automotive	X	X	–	X	75
Aviation	X	X	X	X	100
Medical	X	X	–	X	75
Maritime & energy	X	–	X	X	75
Nuclear	X	–	X	–	50
Railway	X	X	–	X	75
Robotics	X	–	–	–	25
Rate	100%	63%	38%	63%	

Please cite this article in press as: S. Nair et al., An extended systematic literature review on provision of evidence for safety certification, Inform. Softw. Technol. (2014), http://dx.doi.org/10.1016/j.infsof.2014.03.001

Table 7
Challenges and needs addressed in different application domains.

Challenges And Needs	Aerospace	Automotive	Aviation	Medical	Maritime and Energy	Nuclear	Railway	Robotics	Rate (%)
Specification of evidence content	X	X	X	X	X	X	X	X	100
Construction of safety cases	X	X	X	X	X	X	X	X	100
Capturing the degree of credibility or relevance of the evidence	X	X	X	X	–	X	X	–	75
Better development processes and evidence about process compliance	X	X	X	X	X	X	X	–	88
Certification of systems made up of components and subsystems	X	–	X	–	–	–	–	–	25
Ambiguities in safety standards	–	X	X	X	–	X	X	–	63
Demonstration of compliance for novel technologies	X	–	X	–	–	–	–	–	25
Need for providing argumentation	–	–	X	–	–	–	–	–	25
First-time certification or recertification of “proven-in-use” systems	X	–	X	–	–	–	–	–	25
Total	78%	56%	100%	56%	33%	56%	56%	33%	

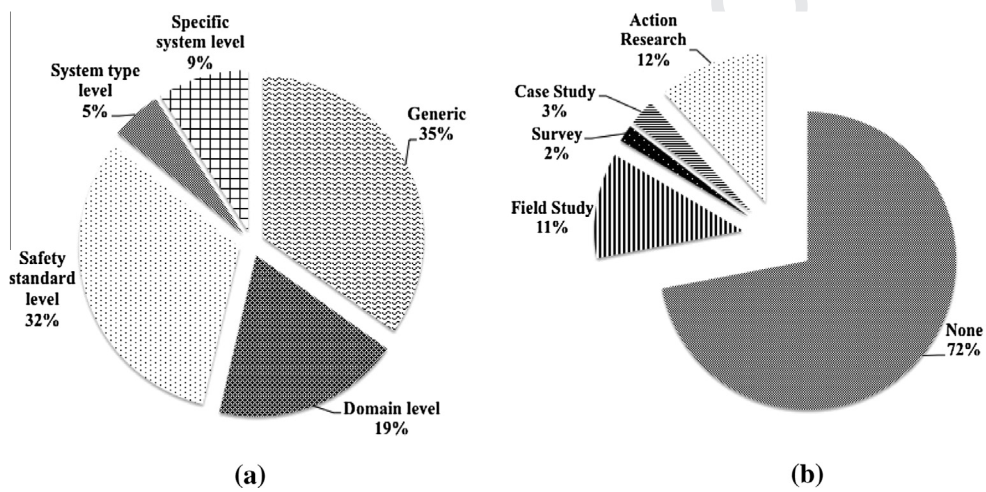

Fig. 5. (a) Percentage of studies for each evidence abstraction level and (b) percentage of studies for each validation method.

e.g. checking consistency and propagation of change in a collection of inter-related evidence artefacts.

An important factor to consider regarding tool support is that safety evidence information is often distributed across different external tools, e.g., requirements management tools, workflow systems, and test automation environments. Consequently, an infrastructure for integration of different (external) tools is necessary. An essential direction to pursue then is providing seamless ways to integrate evidence information originating from different sources. Initiatives such as OSLC (Open Services for Lifecycle Collaboration) [29] can be useful for this purpose. However, several issues must be overcome in order to successfully adopt these frameworks for safety evidence management, such as adequate management of evidence configuration and of evidence granularity [31].

Alongside the taxonomy, our results concerning evidence structuring and assessment serve as useful input for future work on tool support, bringing together and summarising the various techniques that have been proposed for structuring and assessing safety evidence.

For practitioners, the taxonomy can be a helpful tool to gain a clearer understanding of what information may be relevant for demonstration of compliance with safety standards. In particular, information about the evidence types that are validated in real settings or projects can be especially valuable to practitioners. They can benefit from the knowledge assimilated by others from the previous application of the evidence types. In this sense, the specific artefacts, techniques, and tools presented in Appendix B can help practitioners increase their awareness of different alternatives for demonstrating compliance with safety standards.

For most safety standards, some degree of interpretation is required to tailor them to the context of application. In particular, the descriptions provided in safety standards regarding the evidence items are often abstract and in need of interpretation according to contextual factors. In addition to the individual standards being large and requiring interpretation, a system may need to conform to multiple standards. In such cases, it is important to be able to build conceptual relationships between different standards and state how the different evidence items they envisage map onto one another. A taxonomy like the one we have developed is helpful for addressing both of the above problems. First, equipped with the taxonomy, practitioners have a precise and yet concise guide for concepts that are of relevance to safety evidence. This makes it less likely to overlook important information buried in the text of a standard when practitioners are reading and interpreting the standard. Second, the taxonomy can serve as a common framework for mapping the evidence information in different standards. Particularly, one can specify how each standard maps onto the shared taxonomy and use this information to infer and analyse the pairwise relationships between the standards.

Please cite this article in press as: S. Nair et al., An extended systematic literature review on provision of evidence for safety certification, Inform. Softw. Technol. (2014), http://dx.doi.org/10.1016/j.infsof.2014.03.001

Table 8
Number of studies validating each structuring technique, assessment technique and challenge.

	Validated in No. of PS
Evidence structuring technique	
GSN	21
CAE	4
BBN	1
UML models	2
CENELEC templates	1
Trust cases	1
SSG	0
KAOS	2
Structured HTML	0
Structured text	0
Entity-relationship model	2
Process model	2
ACRuDA template	1
Template add-ons	1
Evidence assessment technique	
Argumentation	18
Plain checklists	7
BBN	2
OCL	2
SAL	0
Design checklist	1
GQM-based checklist	2
SEAL	0
Activity-based quality model	0
Evidence-confidence conversion process	0
Taxonomy based questionnaire	0
MODUS	1
Evidence volume approach	0
Challenges and needs identified	
Specification of evidence content	21
Construction of safety cases	15
Capturing the degree of credibility or relevance of the evidence	6
Better development processes and evidence about process compliance	10
Certification of systems made up of components and subsystems	6
Ambiguities in safety standards	4
Demonstration of compliance for novel technologies	3
Need for providing argumentation	2
First-time certification or recertification of "proven-in-use" systems	4

Table 9
Tools identified.

Tools	Validated in No. of PS
ASCE [PS55,PS99,PS10,PS22,PS150,PS173,PS186,PS194,PS5]	9
SAM [PS78,PS164,PS126,PS152,PS183,PS186,PS194,PS215]	8
AutoCERT [PS9,PS42,PS175]	3
Hugin Explorer [PS58,PS167]	2
DECOS test bench [PS3,PS140]	2
VerO-Link analysis tool [PS3]	1
SafeSlice [PS40]	1
LSRD tool [PS79]	1
Unnamed tool based on Ms Excel [PS96]	1
Evidence Agreement tool [PS54]	1
CLawZ toolset [99]	1
TEAMS-RT [PS104]	1
Alloy-based prototype tool [PS116]	1
OSATE [PS56]	1
Unnamed tool [PS11]	1
DOORS/TraceLine [PS45]	1
VAM-LIFE [PS100]	1
Uppaal model checker, AiT tool for Worst case execution time analysis [PS84]	1
RODIN Model prover, ProB tool for model analysis [PS114]	1
Programatica, DevCOP SCMS Eclipse Plug-in [PS142]	1
eSafetyCase Toolkit [PS152]	1
A Markup tool (unnamed) [PS171]	1
SofCheck and GrammaTech [PS93]	1
Extension to Papyrus/Eclipse [PS82]	1
ToolNet [PS131]	1
Excel, Isograph ft+[83]	1
GTO [PS51]	1
Modus [PS137]	1
Unnamed tool [PS148]	1
Visio plug-in for GSN, ASCE [PS174]	1
TCT Editor [PS176]	1
VORD [PS94]	1
An HTML based webpage [PS185]	1
Unnamed tool [PS187]	1
DOVE [PS207]	1
KCG qualified code generator [PS210]	1
Exception analyser [PS213]	1
AdvoCATE [PS43]	1
Objectiver [PS208]	1

Not all the evidence types that we have identified through the review are always required for compliance with a given standard and for a given system. Practitioners will therefore have to determine the types of evidence that they need to provide according to the standards they have to comply with, and in the context of their system or domain. Furthermore, the evidence information has to be agreed upon with a certification authority beforehand. The certifiers may specify additional constraints on the evidence information that needs to be collected. Depending on the regulatory jurisdictions, this may go beyond the requirements stipulated by the standards. In such cases, having a generic taxonomy like the one developed in this paper is beneficial, in the sense that it allows practitioners and certifiers to perform a more thorough analysis of the evidence requirements and reach a consensus about how evidence collection should be carried out.

The taxonomy further provides a common terminology for communication about evidence requirements during the certification process. This helps reduce certification costs by avoiding terminological mismatches. Such mismatches are a common source of problems during certification, arising primarily due to the involvement of multiple experts who have different backgrounds and expertise, and typically different understandings of the evidence required by the safety standards [36].

The results concerned with the evidence taxonomy (RQ1) indicate that the evidence types having to do with safety analysis, requirements, and design have received more attention in the academic literature. This prompts an investigation of the state-of-the-practice to confirm that these types are indeed the most relevant for showing compliance with safety standards. For example, it can be investigated if these types are more frequently used in practice than others such as review results, traceability specification, and functional testing results. Such an investigation will also help in identifying the potential gaps between the state-of-the-art and the state-of-the-practice. Especially, an open issue to investigate is the potential need for further research on the evidence types that were mentioned in only a low percentage of the studies (e.g., *System Testing Results*, *Test Case Specification*). The outcome of such an investigation could be that either: (1) more research is needed to gain insights into the relevance and challenges associated with these types, or; (2) the lack of research is due to practitioners not having recurring problems with these evidence types. Involvement and feedback from the industry would be essential to determine which outcome corresponds to reality.

As indicated by the results in Section 4.6, a large fraction (35%) of the primary studies only had generic-level instances of evidence types. We believe that more research on safety evidence at lower levels of abstraction (system type level and specific system level) is necessary in order to obtain a better understanding of concrete needs and to be able to provide more useful guidance to practitioners.

The results about the type of validation performed in the studies show that the majority (72%) of the studies have not been validated in realistic settings. We view this as a strong indication of the need for work that deals first-hand with the practical aspects of safety certification and provides empirically rigorous analyses of the usefulness of the proposed solutions.

With regards to the tools identified for evidence provision, many of the tools were a combination of prototype verification tools and process management tools to assist with the construction and collection of evidence information. Only 49% of the tools appeared in papers whose results had been validated in real industrial settings. A closer examination of the usefulness and usability of the evidence provision tools in real industrial settings will therefore be an important priority.

The results regarding evidence structuring (RQ2) are useful for both research and practice to promote further work on managing large collections of evidence data. The most widely-identified evidence structuring technique category was *argumentation-induced structuring* (Section4.2), which was validated in 28% of the studies referring to it. To further capitalise on argumentation-induced structuring, future work must focus on effective and modular ways to decompose general safety arguments into coherent and cohesive blocks [28]. This would allow for identifying precisely the evidence required to support each block.

With regards to evidence assessment (RQ3), the most referred to category was *qualitative assessment,* validated in 26% of the studies that referred to it. The results in Section 4.3 indicate that argumentation is the most commonly used technique for qualitative assessment. We believe that to bring about industrial impact in this direction, further research is required to make qualitative reasoning more systematic, particularly when large argumentation structures are involved. Future work must also try to provide automated assistance during evidence assessment to ensure correct execution of the assessment process and the soundness of assessment outcomes. This way, the assessment will become more dependable and less error-prone.

Again, an important remark to make about evidence structuring and assessment is the lack of adequate validation. The large majority of the studies proposing techniques to these ends (63% of structuring and 69% of assessment techniques) were not validated. Similar to the observations made about evidence types and tooling, we believe that more empirical work is required to assess the effectiveness of the proposed structuring and assessment solutions.

With respect to the needs and challenges (RQ4), within the 22-year time window considered, the vast majority of the research (88%) was performed in the last 10 years. To provide a finer-grained analysis of the trends, we show in Fig. 6 the number of papers that tackled each of the identified challenges and needs, distinguishing papers published more than 10 years ago from those published in the last 10 years.

As seen from the figure, *demonstration of compliance for novel technologies* and *first-time certification or recertification of "proven-in-use" systems* have been tackled only in the last 10 years. The emergence of the former challenge may be attributable to the desire to introduce new technologies into safety–critical domains at a faster pace. This could for example be to benefit from technologies that help reduce the carbon-footprint of safety–critical systems and thus ensure that these systems meet the new emission targets and standards that they are subject to. Another motivation could be to facilitate cross-domain reuse, allowing technologies that have a proven track record in their original domain of application to cross over to a new domain (where the technologies would be considered novel) [26]. The emergence of the latter challenge may be attributable to tighter regulations regarding when the proven-in-use clause can be invoked, and also to the increasing demand in the industry for reducing costs [28].

Finally, with regards to the domain analysis of the results (RQ5), we observed that the aviation domain is omnipresent in all aspects of the information gathered. The domain clearly has a leading position on safety certification research and subsequently a large representation in the academic literature. Out of the 218 primary studies identified in the review, 55 were from this domain. A second reason for this large representation is that the aviation domain generally mandates higher bars and a higher level of maturity for safety compliance than others domains. This could mean that some of the evidence types and techniques identified in the aviation domain may be out of scope for other domains. A future analysis of the state-of-the-practice will provide better clues as to which aspects may exclusively concern one domain, e.g., the aviation, but not others.

6. Threats to validity

Following guidelines on validity in SLRs [18], this section discusses the threats to validity of the SLR reported in this paper.

6.1. Publication bias

We began the SLR with limited knowledge about all the related venues. Therefore, we decided to start with an automatic search. After pilot searches, we selected the venues and journals for manual searches. We consider that this mitigated publication bias.

Initially, we did not assume the breadth of the search (i.e., from 1990) and considered as much peer-reviewed literature as possible. Inclusion of grey literature such as PhD theses, technical reports, and whitepapers might have led to more exhaustive results, potentially with a larger representation from the industry. We plan to mitigate this threat in the future by validating the results of the SLR with practitioners. Nevertheless, it is important to note that the inclusion of Google Scholar as a source did not result in the identification of any new evidence type, new category of techniques for evidence structuring and assessment, or new challenges. This makes us believe that the inclusion of grey literature would have little or no effect on the SLR results.

With regards to our search string for automatic search, we avoided, as much as possible, the inclusion of terms that are specific to a certain application domain or a certain technique for demonstration of compliance. However, we were compelled to include in our final search string the terms, safety case, safety argument, assurance case, and dependability case, which are usually associated with the argumentation technique for demonstration of compliance. This decision was in response to an observation made during the pilot searches: there were numerous argumentation-based studies which were concerned with demonstration of compliance to safety standards but which did not explicitly use the term "evidence". This is natural because the presence of evidence is implied in any argumentation structure. Subsequently, the thoroughness of the SLR would have been negatively affected without including these argumentation-related terms in the search string. To mitigate bias towards argumentation techniques, we set stringent requirements in our inclusion criteria, so that a safety argumentation study does not automatically qualify as a primary study but only if it provides insights relevant to safety evidence.

6.2. Selection of primary studies

The first author (PhD candidate) performed most of the selection. This indirectly implies that, due to the lack of adequate expe-

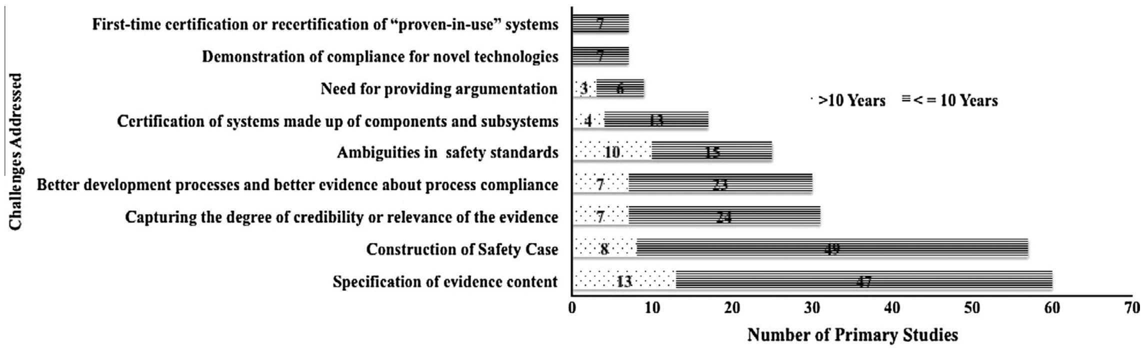

Fig. 6. Compasrison of challenges addressed in the last 10 years with overall challenges identified.

rience or knowledge about the phenomena under study, some publications might have been missed. This is a common threat in SLRs (e.g., [10]), and we performed reliability checks to mitigate it. The reliability checks yielded consistent results with the work of the first author. In addition, well-defined inclusion and exclusion criteria helped reduce researcher bias in the selection of primary studies.

A common threat to the validity of any SLR is the possibility of missing primary studies and thus relevant information. We refined our search string in several iterations, until we were confident that sufficient coverage of literature was obtained. We employed stringent mitigation strategies, including using Google Scholar as an additional source, manual search, reliability checks and expert knowledge, to address this threat to the best of our ability. We believe that the above strategies protect against any major flaws.

The criteria for publication selection (Section3.4) helped us narrow our investigation to a manageable (but still large) size. Although some likelihood exists that relevant studies might have been missed, we consider that the criteria were the best ones given our time and resource constraints. Subsequent studies in the OPENCOSS project[3] [25], e.g., a survey of the project's aviation, railway, and automotive partners about their certification documentation needs [27], have not found any evidence type that is not already included in our proposed taxonomy.

Four primary studies were initially deemed not relevant and excluded during the publication selection process, only to be identified later during the reliability checks. We consider this to be natural because of the broader knowledge gained at Phase 4 of the first publication selection process. The checks were performed at a final stage, after having created a first version of the evidence taxonomy. Therefore, it was easier to identify evidence types, techniques, and challenges. To further mitigate validity threats posed by missing publications, we performed a second publication selection process based on Google Scholar as explained in Section 3.4. The information obtained through this second process did not give rise to any new evidence types, new structuring and assessment techniques, or new challenges. This makes us reasonably confident about the validity of the results reported in the SLR.

6.3. Data extraction and misclassification

In many cases, we had to interpret information and make assumptions about the type of information considered as safety evidence or the validation method used in a study because of the lack of details. The first and the second authors checked, agreed upon, and refined the whole set of data extracted on two occasions in order to mitigate this threat. The validation methods to take into account were also defined before starting data extraction. In relation to the evidence taxonomy, we received feedback on its structure and content from some domain experts.

Finally, although we might have incorrectly extracted and classified some information, we consider that having several studies supporting the definition of each evidence type, technique, and challenge mitigates this threat.

7. Conclusions and future work

Safety certification is a necessary and yet complex activity for most safety–critical systems. One major source of complexity during certification is the specification, collection, and assessment of the evidence required for demonstrating compliance with safety standards. Little has been done in the past to develop a general body of knowledge about safety evidence that is empirically rigours. Motivated by this gap, this paper presented a Systematic Literature Review (SLR) aimed at investigating the state-of-the-art on provision of safety evidence.

One of the main outcomes of the SLR is a general taxonomy of safety evidence types. The taxonomy classifies safety evidence information into 49 basic types (product and process) identified in the literature. We identified that evidence types under *Safety Analysis Results*, *Requirements Specification* and *Design Specification* are the most common in literature.

The SLR further examined and classified existing techniques for structuring evidence information into three categories: *Argumentation-Induced Evidence Structure, Model-Based Evidence Specification,* and *Textual Templates.* Similarly, we classified existing techniques for evidence assessment into four categories: *Qualitative Assessment, Checklists, Quantitative Assessment* and *Logic-based Assessment*.

We also examined the research challenges and needs that have been addressed in the literature. We classified them into nine broad categories and the three most identified referred to the research questions (RQs) of this study: *Specification of evidence content* (RQ1), *Construction of safety cases* (RQ2), and *Capturing the degree of credibility or relevance of the evidence* (RQ3).

Lastly, the paper presented a comparison of eight safety–critical domains in terms of their evidence needs and the relevant challenges. Most information gathered in the review was identified in several domains. In particular, aviation domain was omnipresent in all aspects of the information gathered.

As a major finding, the results about the type of validation performed in the studies indicated that the majority (72%) of the studies have not been validated in realistic settings. We believe that this is a strong indication of the need for more practitioner-oriented and industry-driven empirical studies in the area of safety certification.

The SLR provides useful insights for both researchers and practitioners. From a research standpoint, the evidence taxonomy and the classifications of structuring and assessment techniques pro-

[3] As we stated above, OPENCOSS is the parent project as part of which our SLR was performed.

vide a global overview of existing research on safety evidence. This is helpful both as a general introduction to the area, and also as a reference for organising future research. The challenges and needs that have been identified are useful for developing a future research agenda.

As for practitioners, the results, particularly the evidence taxonomy developed, provide a concrete reference for learning and tailoring the various types of evidence that may be required during certification. Moreover, the taxonomy creates a common terminology for safety evidence. Having such a common terminology is advantageous both as a vehicle to facilitate communication and avoid misunderstandings, and also as a basis around which tool support can be designed for safety evidence management. Requirements for such tool support can be elicited from the results of the SLR. Among them, integration with other tools seems to be a key aspect to address.

The SLR is part of a larger and on-going research effort aimed at improving safety certification practices. We emphasise that the SLR is focused exclusively on academic literature. Subsequently, no conclusions can be drawn based on our current results by way of correlating the proportional number of studies on a certain technique and the usefulness of the technique in practice. Analysing practical usefulness and industrial adoption requires studies on the current state of practice and is outside the scope of this SLR.

In the future, we would like to further analyse the dependencies and constraints between different evidence types and create more detailed models of evidence information in different domains. To further ground our the results of the SLR in industrial needs, we plan to validate the findings of the review by (1) conducting new empirical studies (e.g., surveys) for investigating how practitioners provide evidence for safety certification and (2) comparing the evidence taxonomy developed, together with its glossary, to the information presented in different safety standards regarding the evidence to provide to comply with them. These studies would allow us to compare the state of the art and the state of the practice, in relation to both what practitioners do and what safety standards indicate. We could also compare how different evidence types of the taxonomy (i.e., notions of information that constitute safety evidence) are referred to and defined in different application domains, determining their differences and commonalities. This would also allow us to find the notions with which some confusion or discrepancies exist among different application domains.

8. Uncited references

[PS2,PS4,PS13,PS15–PS18,PS21,PS24–PS29,PS31,PS33–PS36,PS39,PS44,PS48,PS49,PS59–PS61,PS63–PS65,PS68,PS70,PS73–PS77,PS81,PS85–PS92,PS95,PS97,PS102,PS103,PS105–PS108,PS110,PS112,PS113,PS115,PS117,PS119,PS120,PS123,PS125,PS127,PS129,PS130,PS132,PS133,PS135,PS141,PS143,PS144,PS146,PS147,PS151,PS153–PS156,PS158,PS160,PS161,PS163,PS165,PS166,PS169,PS177,PS179–PS182,PS184,PS188–PS193,PS195–PS206,PS209,PS211,PS214,PS216–PS218].

Acknowledgments

The research leading to this paper has received funding from the FP7 programme under the Grant Agreement No. 289011 (OPENCOSS), from the National Research Fund, Luxembourg (FNR/P10/03 – Validation and Verification Laboratory), and from the Research Council of Norway under Certus SFI. We would like to thank Shaukat Ali, Laurent de la Beaujardiere, Arnaud Gotlieb, Oleg Lisagor, Shiva Nejati, and Paolo Panaroni for their useful feedback on the evidence taxonomy. We are also grateful to the reviewers for their valuable suggestions towards improving the paper.

Appendix A. Examples of data extracted from the primary studies

Bibliographic information	Application domain(s)	Underlying standard(s)	Information/artefact/tool/technique contributing to evidence	Techniques for evidence structuring	Techniques for assessing evidence confidence	Tool support	Objectives/challenges addressed	Evidence abstraction level	Validation method
Andersen, B.S., Romanski, G [PS3] ACM Digital Library [2011]	Aviation	DO-178B	Structural Coverage Analysis, Plan For Software Aspects Of Certification, Software Development Plan, Software Verification Plan, Software Configuration Management Plan, Quality Assurance Plan, Software Design Artefacts, Source Code, Verification Methods And Data, Formal Methods, Exhaustive Input Testing, Structural Coverage Analysis Report And Its Review Checklist	None	None	VerO-Link Analysis tool	Better development processes and better evidence about process compliance	System type level, Standard Level	None
Arthasartsri, S., Ren, H [PS6] IEEE	Aviation	Unspecified	Functional Hazard Analysis, Preliminary Risk Analysis, CMA,	None	None	None	Better development	Domain and Specific	Case Study

(continued on next page)

Appendix A:Examples Of Data Extracted From The Primary Studiesof data extracted from the primary studies (*continued*)

Bibliographic information	Application domain(s)	Underlying standard(s)	Information/artefact/tool/ technique contributing to evidence	Techniques for evidence structuring	Techniques for assessing evidence confidence	Tool support	Objectives/ challenges addressed	Evidence abstraction level	Validation method
[2009]			HHA, FHA, IHA, ECHA, RASP, CMA, MMEL/CDL, FMEA, FMES, Safety Assessment Reliability Prediction, Equipment Cmas				processes and better evidence about process compliance	System Level	
Linling, S. Kelly, T. [PS100] IEEE [2009]	Aviation	Unspecified	Simulation, Historical Service Data, Design Rules, FTA	GSN, CAE	Argumentation	VAM-LIFE	Need for providing argumentation	Domain Level	None
Graydon, P., Habli, I., Hawkins, R., Kelly, T., Knight, J [PS174] Expert Knowledge [2011]	Aviation	DO-178B	Operating System, Code Review, Code Inspection, Branch Coverage Testing, Test Plan, Boundary Values Testing, Test Case Specification	GSN Models & CAE	Argumentation	Visio Plugin for GSN, CAE	Capturing the degree of credibility or relevance of the evidence	Specific System Level	Action Research
L.H. Eriksson [PS51] Safecomp [2004]	Railway	EN50126, EN50128, and EN50129	System Definition (Design Documentation), Quality Management Report, Safety Management Report, Technical Safety Report, Related Safety Cases, Installation Structure, Automated Theorem Proving (In Propositional Logic), Risk Analysis	CENELEC template	Checklist	GTO	Construction of Safety cases	Safety Standard Level, Specific system Level	Action Research, Survey
S. Wagner, B. Schatz, S. Puchner, P. Kock [PS159] IEEE [2010]	Automotive	IEC61508	Use Of Fault Pattern Libraries (Source Code), Testing Results Using Fault Injection, Formal Verification Results, Simulation, Fault Models (Hazard Analysis), Simulink / Stateflow / Targetlink Models	GSN Models	Argumentation	None	Ambiguity in Safety Standards	Domain Level, System Type Level	Case Study
J. Wang. [PS212] Google Scholar [2000]	Energy and Oil	Multi standards	FTA, Consequence Analysis, ETA, Structural Review Of Risks, Requirements Analysis, Safety Requirements Specifications, Systematic Audit To Confirm The Safety Requirements Specifications Meets Software, Semantic Analysis, Software Reliability Growth Models, Formal Methods Like Z; Vienna Development Method, Communicating Sequential Processes And Calculus Of Communicating System, FMECA, PHA	None	None	None	Specification of evidence content	Domain Level	None

Appendix A:Examples Of Data Extracted From The Primary Studiesof data extracted from the primary studies (*continued*)

Bibliographic information	Application domain(s)	Underlying standard(s)	Information/artefact/tool/technique contributing to evidence	Techniques for evidence structuring	Techniques for assessing evidence confidence	Tool support	Objectives/challenges addressed	Evidence abstraction level	Validation method
Stephenson, Z., Fairburn, C., Despotou, G., Kelly, T., Herbert, N., Daughtrey, B [PS149] Springer [2011]	Unspecified	Unspecified	HAZOP; FTPC; FFA; FMEA; HEP; HRA	None	None	None	Certification of systems made up of components and subsystems, Construction of safety cases	Generic	None
Valk, J.-L., Vis, H., & Koning, G. Phileas [PS157] Springer [2010]	Railway	CENELEC	PHA, FTA, hazard log, safety requirements, traceability of the requirements flow down, architectural design, Independent Verification and Validation, Quality assurance of the development process, requirements traceability between models and formal requirements, Review and static analysis at the model level to guarantee compliance to modelling standards, Functional verification of the models by using requirements based test vectors, Automatic code generation with built in traceability between the source code and the models, Code review, Equivalence testing, System Requirements Specification; safety Requirements Specification, Safety Assessment Report	None	None	None	Ambiguity in Safety Standards, Specification of evidence content	Safety Standard Level and Specific System Level	None
Hamilton, V [PS72] Springer [2011]	Unspecified	DO-178B, IEC 61508	Safety management plan, software development and verification plans, HAZOP, software design specification, integration test results, static analysis of code, design reviews, normal range testing, traceability specification	GSN, CAE	Argumentation	None	Specification of evidence content	Generic	None

Please cite this article in press as: S. Nair et al., An extended systematic literature review on provision of evidence for safety certification, Inform. Softw. Technol. (2014), http://dx.doi.org/10.1016/j.infsof.2014.03.001

Appendix B. Glossary of evidence types

We need to make the following clarifications to ensure a better understanding of the taxonomy and how it was built:

– After finding information that could be regarded as evidence in the publications, we classified it in different categories.

– From a (business) process perspective [5]:
  - The tasks related to building, maintaining and using a critical system are specified in the *Activity Planning*.
  - The roles that will execute the tasks are specified in the *Activity Planning*.
  - The skills and knowledge required (conditions) for task execution are specified in *Personnel Competence*.
  - The necessary inputs (which exist before the critical system is built) correspond to *Tool Support* and *Reused Components Information*.
  - The outputs (i.e., results) of the process correspond to *Activity Records* and *Product Information*.
  - The output of one task can be input for another.

– *Product Information* also corresponds to *Activity Records* (i.e., product information shows the activities performed).

– We found that *Historical Service Data* can refer both to a component that will be reused in a new system and to an existing system that aims to be (re-)certified after having been in operation. We have considered that the same techniques, artefacts, and information can be used for the evidence types defined for both cases (*Reused Component Historical Service Data Specification* and *System Historical Service Data Specification*).

– The structure of *Safety Analysis Results* is based on the common explanation and relationships between accidents (aka mishaps), risks, and hazards (e.g., [9]).

– Many techniques for safety analysis can be used to specify several types of evidence. For example, FTA can be used for Hazard Cause Specification and Risk Analysis Results [9].

– The information regarding static analysis, inspections, and reviews indicated in the studies of the SLR has only been considered relevant if the publications indicated the element (i.e., artefact) under analysis (e.g., "source code static analysis").

– *Test Cases Specification* can refer to any type of Testing Results (e.g., unit test cases). These types have only been included in *Testing Results* to minimise the size of the taxonomy.

– The structure of the child nodes of Testing Results is based on the testing types classification presented in [1].

– There exist relationships and constraints between evidence types. For example, certain *Testing Results* are linked to the *Requirements Specification*. They are currently not specified in the taxonomy.

– When specifying test cases and providing test results, a combination of *target-based testing, objective-based testing,* and *environment-based testing* can be used (e.g., system-performance-operational testing).

The following table presents a glossary to support the understanding of the Taxonomy (Fig. 2) with information such as definition of each evidence type, information, techniques, tools and artefacts extracted and classified accordingly from the primary studies.

Acceptance testing results
Definition: Results from the validation of the behaviour of a critical system against its customers' requirements. The customers undertake or specify typical tasks to check that their requirements have been met [1]
Techniques: user evaluation in mock work environments

Accidents Specification
Definition: Specification of the events that result in an outcome culminating in death, injury, damage, harm, and/or loss as a consequence of the occurrence of a hazard of a critical system [9]
Techniques: ETA; PHL; PHA; FMEA; FMECA; FMES; IHA; FMEDA

Activity Records
Definition: Specification of the work performed to execute the activity planning of a critical system [9]
Artefacts: QA audit results; maintenance log; change requests report; system changes report; review checklists; quality management report; safety management report; technical safety report; risk management file; safety and engineering meeting minutes; design checklists; V&V effort report; configuration control records; QA activities report; quality control documents; safety criteria report; safety compliance assessment report; failure checklist; customer feedback reports; feasibility analysis; implementation track; integration report; quality management report; project execution report; hazard checklist; report on monitoring operator performance and periodic review of skills; structural coverage analysis review checklist; SAS
Information: testing team independence

Architecture Specification
Definition: Description of the fundamental organisation of a critical system, embodied in its components, their relationships to each other and to the environment, and the principles guiding its design and evolution [15]
Technique: AADL
Artefacts: dependence diagram

Assumptions and Conditions Specification
Definition: Description of the constraints on the working environment of a critical system for which it was designed [35]
Artefacts: assumptions about the environment where the code is executed; domain assumptions

Automated Static Analysis Results
Definition: Results from an automatic process for evaluating a critical system based on its form, structure, content, or documentation [32]
Techniques: code static analysis; fault model static analysis; control flow analysis; worst case execution time analysis; integrity analysis; cyclomatic complexity analysis; data coupling analysis; control coupling analysis

Communication Plan
Definition: Description of the activities targeted at creating project-wide awareness and involvement in the development of a critical system [9]

Please cite this article in press as: S. Nair et al., An extended systematic literature review on provision of evidence for safety certification, Inform. Softw. Technol. (2014), http://dx.doi.org/10.1016/j.infsof.2014.03.001

Configuration Management Plan
Definition: Description of how identification, change control, status accounting, audit, and interface of a critical system will be governed [5][4]
Artefacts: SCMP; version management; change control procedures.
Information: target platform

Design Specification
Definition: Specification of the components, interfaces, and other internal characteristics of a critical system or component [5][32]
Techniques: ADDL; UML; SysML; SCADE.
Artefacts: Interface design; data structures; state machine.
Information: safety assessment reliability prediction.

Development Plan
Definition: Description of how a critical system will be built. It includes information about the requirements, design, and implementation (coding and/or integration) phases [5]
Artefacts: SDP; test generation procedure; verification process
Information: Development methodology; coding standards; coding guidelines; design rules; pair-programming; use of industry-standard state machine notations; metrics for function-code size; FFPA method; design technique; implementation technique

Development and V&V Staff Competence Specification
Definition: Specification of the skills or knowledge that the parties involved in the development and V&V plans of a critical system need in order to carry out the activities assigned to them [35]
Artefacts: developer qualification; engineers CV
Information: Staff experience; authority and training; tool training; software architects experience; experience, authority, and training of verification engineers; reviewer competence

Functional Testing Results
Definition: Results from the validation of whether or not the observed behaviour of a system conforms to its specification [1]
Techniques: hazard directed testing

Hazards Causes Specification
Definition: Specification of the factors that create the hazards of a critical system [9]
Techniques: FTA; FMEA; FMECA; anthropometric and workload assessment; Markov Analysis; HAZOP; causal analysis; SHARD; common failure analysis; common mode failure analysis; common mode analysis; root cause analysis; FMES; FPTC; FPTN; IHA; FFA; ECHA; HEP; HRA; FMEDA
Information: human error

Hazards Specification
Definition: Specification of the conditions in a critical system that can become a unique, potential accident [9]
Techniques: PHL; PHA; SHA; HHA; FMEA; FMECA; FHA; Petri Nets; Markov Analysis; HAZOP; SHARD; HAZID; FMES; vulnerability analysis; IHA; ECHA; HEP; HRA FMEDA
Artefacts: hazard log

Hazards Mitigation Specification
Definition: Specification of how to reduce hazard likelihood and hazard consequences when a hazard cannot be eliminated in a critical system [9]
Synonyms: hazard contingency specification, hazard barriers specification, and hazard protections specification
Techniques: PHA; SHA; FMECA; IHA; ECHA; diversity analysis; FMEDA

Historical Service Data Specification
Definition: Specification of the dependability (often, reliability) of a component reused in a critical system based on past observation of the behaviour of the component [35]
Artefacts: field service experience; product service history; fault log; maintenance reports; studies and reviews of operation safety and environmental experience; maintenance records and surveys
Information: probability of failure on demand (from past behaviour); prior field reliability in similar applications; failure frequency; failure rate; MTTF; MTTR; MTBF

Inspection Results
Definition: Results from the visual examination of system lifecycle work products of a critical system to detect errors, violations of development standards, and other problems [32]
Synonyms: audit (usually used to refer to inspections made by an independent party [32]
Technique: functional configuration audit; physical configuration audit; inspection of safety requirements; code inspection; independent analysis of requirements and architecture specification; safety audit; independent assessment of tests
Artefacts: independent safety audit report

Integration Testing Results
Definition: Results from the evaluation of the interaction between the components of a system [1]
Techniques: software integration testing; hardware integration testing; interfaces testing

Model Checking Results
Definition: Results from the verification of the conformance of a critical system to a given specification by providing a formal guarantee. The critical system under verification is modelled as a state transition system, and the specifications are expressed as temporal logic formulae that express constraints over the system dynamics [5]
Techniques: CCS; CSP; LOTOS; temporal logic; Lustre; ASA; ClawZ; Uppaal; lambda calculus; schedulability analysis; Time Petri Nets.
Tools: Uppaal

Modification Procedures Plan
Synonyms: maintenance procedures plan
Definition: Description of the instructions as to what to do when performing a modification in a critical system in order to make corrections, enhancements, or adaptations to the validated system, ensuring that the required safety is sustained [35]
Techniques, tools and artefacts: changes propagation; non-regression testing; maintenance plan; inspection procedures; repair time; change assessment

Non-operational Testing Results
Definition: Results from evaluation of a critical system in an environment that does not correspond to but replicates its actual operational environment [1]

Normal Range Testing Results
Definition: Results from the verification of the behaviour of a system under normal operational conditions [13]

(continued on next page)

Techniques: Equivalence classes and input partitioning testing.

Object Code
Definition: Computer instructions and data definitions in a form output by an assembler or compiler [32]

Operation Procedures Plan
Definition: Description of the instructions and manuals necessary to ensure that the safety targets of a critical system are maintained during its use [35]
Artefacts: user manual; target staff description; installation procedure; operational staff support description; installation structure plan; training plan; incident registration procedures; performance monitoring plan; installation and operation facility procedures; evacuation procedures; description of the allocation of system functions between equipment and operators

Operational Testing Results
Definition: Results from the evaluation of a critical system in its actual operating environment [1]

Operator Competence Specification
Definition: Specification of the skills or knowledge that the parties involved in the operation procedures need in order to carry out the activities assigned to them [35]
Techniques, tools and artefacts: operational staff training needs specification; manning requirements specification.
Information: operator competence; user experience.

Performance Testing Results
Definition: Results from the verification of the performance requirements (e.g., capacity and response time) of a critical system [1]
Synonyms: resource consumption analysis
Techniques: memory use analysis; timing analysis; memory partitioning analysis
Information: memory use

Project Risk Management Plan
Definition: Description of the activity regarding the development and documentation of an organised and comprehensive strategy for identifying project risks. It includes establishing methods for mitigating and tracking risk [9]

Reliability Testing Results
Definition: Results from the verification of fault-free behaviour in a critical system [1]
Synonyms: failure analysis
Techniques: statistical testing; probabilistic testing

Requirements Specification
Definition: Specification of the external conditions and capabilities that a critical system must meet and possess, respectively, in order to (1) allow a user to solve a problem or achieve an objective, or (2) satisfy a contract, standard, specification, or other formally imposed documents [5][32]
Artefacts: (specifications of) performance requirements; derived requirements; software safety requirements; software requirements; high-level requirements; low-level requirements; functional requirements; interface requirements; safety requirements; failure requirements; monitoring requirements; software requirements; MMEL/CDL

Reused Component Specification
Definition: Specification of the characteristics of an existing system that is (re-) used to make up a critical system [32]
Artefacts: reused component requirements specification; reused component functions specification; fault pattern library; reused component reliability specification; product safety accreditation; OS/RTOS certification; supplier information; reused component safety case; reused component safety analysis results; equipment requirements specification

Reused Component Historical Service Data Specification
Definition: Specification of the dependability (often, reliability) of a component reused in a critical system based on past observation of the behaviour [35]
Artefacts: field service experience; product service history; fault log; maintenance reports; studies and reviews of operation safety and environmental experience; maintenance records and surveys
Information: probability of failure on demand (from past behaviour); prior field reliability in similar applications; failure frequency; failure rate; MTTF; MTTR; MTBF

Review Results
Definition: Description of a process or meeting during which a system lifecycle work product or set of works products is presented to some interested party for comment or approval [32].
Synonyms: walkthrough (usually used to refer to a review led by a designer or programmer)
Artefacts: (results from, usually reports of) source code walkthrough; independent audit review; source code review; design review

Risk Analysis Results
Definition: Specification of the expected amount of danger when an identified hazard will be activated and thus become an accident in a critical system [9]
Synonyms: risk assessment results
Techniques: FTA; ETA; PHA; SHA; FMEA; FMECA; Markov Analysis; FMES; FPTC; FPTN; PHA; FMES; IHA; RASP; HRA
Information: likelihood, severity

Project Risk Management Plan
Definition: Description of the activity regarding the development and documentation of an organised and comprehensive strategy for identifying project risks. It includes establishing methods for mitigating risk and for tracking risk [9]

Robustness Testing Results
Definition: Results from the verification of the behaviour of a critical system in the presence of faulty situations in its environment [1]
Techniques: fault injection testing; SWIFI; EMFI

Safety Management Plan
Definition: Description of the coordinated, comprehensive set of processes designed to direct and control resources to optimally manage the safety of an operational aspect of an organisation [9]

Simulation Results
Definition: Results from the verification of a critical system by creating a model that behaves or operates like the system when provided with a set of controlled inputs [32]
Techniques: symbolic execution; emulation; hardware-in-loop testing; animation
Tools: Matlab/Simulink; TargetLink; Stateflow

Source Code

Please cite this article in press as: S. Nair et al., An extended systematic literature review on provision of evidence for safety certification, Inform. Softw. Technol. (2014), http://dx.doi.org/10.1016/j.infsof.2014.03.001

Definition: Computer instructions and data definitions expressed in a form suitable for input to an assembler, compiler, or other translator [32].
Artefacts: ADA code; C code; C++ code

Stress Testing Results
Definition: Results from the verification of the behaviour of a critical system at the maximum design load, as well as beyond it [1]
Techniques: boundary value testing; exhaustive input testing; sensitivity testing

Structural Coverage Testing Results
Definition: Results from the verification of the behaviour of a critical system by executing all or a percentage of the statements or blocks of statements in a program, or specified combinations of them, according to some criteria [1]
Synonyms: structural coverage analysis
Techniques: MC/DC testing (or coverage); control flow analysis; data flow analysis; statement coverage; branch coverage; subroutines coverage; safety requirements coverage
Information: element under analysis; coverage percentage

System Historical Service Data Specification
Definition: Specification of the dependability (often, reliability) of a system based on past (prior-certification) observation of the behaviour [35]

System Inception Specification
Definition: Specification of initial details about the characteristics of a critical system and how it will be created [5][13]
Artefacts: PSAC; EUC specification; scoping document
Information: suitability of notations; soundness of methods; quality of development method

References

[1] A. Abran, J.W. Moore, Guide to the software engineering body of knowledge, IEEE Comput. Soc. (2004).
[2] W. Afzal, R. Torkar, R. Feldt, A systematic review of search-based testing for non-functional system properties, Inf. Softw. Technol. 51 (2009) 957–976.
[3] P. Baufreton, J. Blanquart, J. Boulanger, H. Delseny, J. Derrien, J. Gassino, G. Ladier, E. Ledinot, M. Leeman, P. Quéré, Multi-domain comparison of safety standards, in: Proceedings of the 5th International Conference on Embedded Real Time Software and Systems (ERTS2 2010), Toulouse, France (May 19–21, 2010), 2010.
[4] H.R. Berlack, Software Configuration Management, Wiley Online Library, 1992.
[5] M. Bozzano, A. Villafiorita, Design and Safety Assessment of Critical Systems, Auerbach Pub, 2010.
[6] G. Caldiera, B. Victor, H.D. Rombach, The goal question metric approach, Encyclopedia Softw. Eng. 2 (1994) 528–532.
[7] DO-178C/ED-12C, Software Considerations in Airborne Systems and Equipment Certification, 2012.
[8] Draft International Standard Road vehicles—Functional safety – ISO/DIS 26262–8, 2009.
[9] C.A. Ericson, Concise Encyclopedia of System Safety: Definition of Terms and Concepts, Wiley, Hoboken, 2011.
[10] A. Fernandez, E. Insfran, S. Abrahão, Usability evaluation methods for the web: a systematic mapping study, Inf. Softw. Technol. 53 (2011) 789–817.
[11] Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems (IEC 61508), 2005.
[12] M. Gerlach, R. Hilbrich, S. Weißleder, Can cars fly? From avionics to automotive: comparability of domain specific safety standards, in: Proceedings of the Embedded World Conference, 2011.
[13] V. Hilderman, T. Baghi, Avionics certification: a complete guide to DO-178 (software), DO-254 (hardware), Avionics Communications, 2007.
[14] J. Hill, D. Victor, The product engineering class in the software safety risk taxonomy for building safety-critical systems, in: Proceedings of the 19th Australian Software Engineering Conference (ASWEC), IEEE CS Press, March 2008, pp. 617–626.
[15] B. Hjørland, H. Albrechtsen, Toward a new horizon in information science: domain-analysis, JASIS 46 (6) (1995) 400–425.
[16] D. Jackson, M. Thomas, L.I. Millett, Software for Dependable Systems: Sufficient Evidence?, National Academies Press, 2007
[17] T.P. Kelly, Arguing Safety – A Systematic Approach to Managing Safety Cases, PhD thesis, University of York, 1998.
[18] B.A. Kitchenham, S. Charters, Guidelines for Performing Systematic Literature Reviews in Software Engineering Version 2.3, EBSE Technical Report, Keele University and University of Durham, 2007.
[19] Ministry of Defence, Defence Standard 00–56 Issue 4: Safety Management Requirements for Defence Systems, 2007.
[20] S. Nair, J. delavara, M. Sabetzadeh, L. Briand, SLR on Evidence Classification, Structuring and Assessment for Safety: Extracted Data, Techical Report, 2012. <https://simula.no/publications/SLR_Full_Extracted_Data/simula_pdf_file>.
[21] S. Nair, J. delavara, M. Sabetzadeh, L. Briand, Classification, structuring and assessment of evidence for safety – a systematic literature review, in: Sixth IEEE International Conference on Software Testing, Verification and Validation (ICST), 2013.
[22] J. Nicolás, A. Toval, On the generation of requirements specifications from software engineering models: a systematic literature review, Inf. Softw. Technol. 51 (2009) 1291–1307.
[23] Object Management Group (OMG), OMG Object Constraint Language, 2006. <http://www.omg.org/spec/OCL/2.0/>.
[24] OMG, Software Assurance Case Metamodel (SACM), 2013.
[25] OPENCOSS. <http://opencoss-project.eu> (accessed 20.05.12).
[26] OPENCOSS. D1.2 – Industrial Use Cases: Description and Business Impact, 2012.
[27] OPENCOSS. D6.1 – Baseline for the Evidence Management Needs, 2012.
[28] OPENCOSS. D5.1 – Baseline for the Compositional Certification Approach, 2012.
[29] Open Services for Lifecycle Collaboration (OSLC), 2013. <http://open-services.net>.
[30] Oxford Dictionaries, Evidence, <http://oxforddictionaries.com/definition/evidence?q=evidence> (accessed 25.05.12).
[31] Proposal Towards adaptation of OSLC – iFEST, 2013. <http://www.artemis-ifest.eu/node/45>.
[32] J. Radatz, A. Geraci, F. Katki, IEEE standard glossary of software engineering terminology, IEEE Std. 610121990 (1990) 121990.
[33] Railway Applications – Safety Related Electronic Systems for Signalling, European Committee for Electrotechnical Standardisation CENELEC ENV 50129, 1998.
[34] P. Rodríguez-Dapena, Software safety certification: a multidomain problem, IEEE Softw. 16 (1999) 31–38.
[35] M. Sabetzadeh, D. Falessi, L. Briand, R. Panesar-Walawege, C. Markusen, R. Morgan, J. Borg, T. Coq, MODUS: A goal-based approach for quantitative assessment of technical systems. ModelME! Technical Report (accessed 25.05.12).
[36] M. Sabetzadeh, S. Nejati, L. Briand, A.E. Mills, Using SysML for modeling of safety-critical software-hardware interfaces: Guidelines and industry experience, in: 13th International Symposium on High-Assurance Systems Engineering (HASE), 2011, pp. 193–201.
[37] A. Singhal, A. Singhal, A systematic review of software reliability studies, Softw. Eng. Int. J. 1 (1) (2011).

Primary Studies

[PS1] W. Alan, M. Tom, L. Mark, B. Hans, Software certification: is there a case against safety cases? in: Foundations of Computer Software, Modelling, Development, and Verification of Adaptive Systems, Springer, 2011, pp. 206–227.
[PS2] B.S. Andersen, G. Romanski, Verification of safety-critical software, Commun. ACM 54 (2011) 52–57.
[PS3] E. Althammer, E. Schoitsch, G. Sonneck, H. Eriksson, J. Vinter, Modular certification support—the DECOS concept of generic safety cases, in: 6th International Conference on Industrial Informatics (INDIN), 2008, pp. 258–263.
[PS4] K.J. Anderson, Common law safety case approaches to safety critical systems assurance, in: Developments in Risk-based Approaches to Safety, Springer, 2006, pp. 171–183.
[PS5] T.S. Ankrum, A.H. Kromholz, Structured assurance cases: three common standards, in: 9th International Symposium on High-Assurance Systems Engineering (HASE), 2005, pp. 99–108.
[PS6] S. Arthasartsri, H. Ren, Validation and verification methodologies in A380 aircraft reliability program, in: 8th International Conference on Reliability, Maintainability and Safety (ICRMS), 2009, pp. 1356–1363.
[PS7] A. Bain, S. Dobson, Safety cases for legacy warships: a systematic approach, in: 3rd IET International Conference on System Safety, 2008, pp. 1–6.
[PS8] N. Basir, E. Denney, B. Fischer, Constructing a safety case for automatically generated code from formal program verification information, in: Computer Safety, Reliability, and Security, Springer, 2008, pp. 249–262.
[PS9] N. Basir, E. Denney, B. Fischer, Deriving safety cases for hierarchical structure in model-based development, in: Computer Safety, Reliability, and Security, Springer, 2010, pp. 68–81.
[PS10] N. Basir, E. Denney, B. Fischer, Deriving safety cases for the formal safety certification of automatically generated code, Electron. Notes Theor. Comput. Sci. 238 (2009) 19–26.

[PS11] N. Basir, E. Denney, B. Fischer, Deriving safety cases from automatically constructed proofs, in: 4th IET International Conference on Systems Safety, 2009, 14–14.
[PS12] I. Bate, P. Conmy, J. McDermid, Generating evidence for certification of modern processors for use in safety-critical systems, in: 5th International Symposium on High Assurance Systems Engineering (HASE), 2000, pp. 125–134.
[PS13] I. Bate, P. Conmy, Certification of FPGAs-current issues and possible solutions, in: Safety-Critical Systems: Problems, Process and Practice, Springer, 2009, pp. 149–165.
[PS14] I. Bate, T. Kelly, Architectural considerations in the certification of modular systems, Reliab. Eng. Syst. Safety 81 (2003) 303–324.
[PS15] U. Becker, Applying safety goals to a new intensive care workstation system, in: Computer Safety, Reliability, and Security, Springer, 2008, pp. 263–276.
[PS16] O. Benediktsson, R. Hunter, A.D. McGettrick, Processes for software in safety critical systems, Softw. Process: Improve. Practice 6 (2001) 47–62.
[PS17] A.F. Benet, A risk driven approach to testing medical device software, in: Advances in Systems Safety, Springer, 2011, pp. 157–168.
[PS18] A. Bertolino, L. Strigini, Assessing the risk due to software faults: estimates of failure rate versus evidence of perfection, Softw. Test., Verif. Reliab. 8 (1998) 155–166.
[PS19] C.G. Bilich, Z. Hu, Experiences with the certification of a generic functional safety management structure according to IEC 61508, in: Computer Safety, Reliability, and Security, Springer, 2009, pp. 103–117.
[PS20] P. Bishop, R. Bloomfield, A methodology for safety case development, in: Industrial Perspectives of Safety-critical Systems, Springer, 1998, pp. 194–203.
[PS21] P. Bishop, R. Bloomfield, B. Littlewood, A. Povyakalo, D. Wright, Toward a formalism for conservative claims about the dependability of software-based systems, IEEE Trans. Softw. Eng. 37 (2011) 708–717.
[PS22] R. Bloomfield, P. Bishop, Safety and assurance cases: Past, present and possible future–an Adelard perspective, in: Making Systems Safer, Springer, 2010, pp. 51–67.
[PS23] M. Bouissou, F. Martin, A. Ourghanlian, Assessment of a safety-critical system including software: a Bayesian Belief Network for evidence sources, in: Reliability and Maintainability Symposium, Proceedings. Annual, IEEE, 1999, pp. 142–150.
[PS24] A. Brown, J. Fenn, C. Menon, Issues and considerations for a modular safety certification approach in a Service-Oriented Architecture, in: 5th IET International Conference on System Safety, 2010, pp. 1–6.
[PS25] M. Brown, Rationale for the development of the UK defence standards for safety-critical computer software, Aerospace Electron. Syst. Mag., IEEE 5 (1990) 31–37.
[PS26] A. Burns, J. McDermid, Real-time safety-critical systems: analysis and synthesis, Softw. Eng. J. 9 (1994) 267–281.
[PS27] J.L. Camus, Efficient development of safety-critical software, Electron. Syst. Softw. 1 (2003) 38–43.
[PS28] P. Caseley, M. Hadley, Assessing the effectiveness of static code analysis, in: 1st IET International Conference on System Safety, 2006, pp. 227–237.
[PS29] P. Caseley, T. White, The MOD procurement guidance on software safety assurance-assessing and understanding software evidence, in: Incorporating the SaRS Annual Conference, 4th IET International Conference on Systems Safety, 2009, pp. 1–12.
[PS30] P. Chinneck, D. Pumfrey, J. McDermid, The HEAT/ACT preliminary safety case: a case study in the use of goal structuring notation, in: Proceedings of the 9th Australian workshop on Safety critical systems and software-Volume 47, Australian Computer Society Inc, 2004, pp. 33–41.
[PS31] T. Cichocki, Safety case development how can I continue the work? in: Improvements in System Safety, Springer, 2008, pp. 59–76.
[PS32] J. Clegg, Arguing the safety of FPGAs within safety critical systems, in: 4th IET International Conference on Systems Safety, 2009, 52–52.
[PS33] P. Conmy, I. Bate, Component-based safety analysis of FPGAs, IEEE Trans. Ind. Informat. 6 (2010) 195–205.
[PS34] P. Conmy, R.F. Paige, Challenges when using model driven architecture in the development of safety critical software, in: 4th International Workshop on Model-Based Methodologies for Pervasive and Embedded Software (MOMPES), 2007, pp. 127–136.
[PS35] J.D. Corrie, Safety assurance and safety assessment, in: 11th IET International Conference on Systems Safety, Railway Signalling and Control Systems, 2006, pp. 29–46.
[PS36] C. Cruz-Neira, R.R. Lutz, Using immersive virtual environments for certification, IEEE Softw. 16 (1999) 26–30.
[PS37] B.J. Czerny, J.G. D'Ambrosio, B.T. Murray, Providing convincing evidence of safety in X-by-wire automotive systems, in: 5th International Symposium on High Assurance Systems Engineering (HASE), 2000, pp. 189–192.
[PS38] G. Dahll, Combining disparate sources of information in the safety assessment of software-based systems, Nucl. Eng. Des. 195 (2000) 307–319.
[PS39] D. Schneider, M. Trapp, Conditional safety certificates in open systems, in: Proceedings of the 1st Workshop on Critical Automotive Applications: Robustness & Safety, ACM, 2010, pp. 57–60.
[PS40] D. Falessi, S. Nejati, M. Sabetzadeh, L.C. Briand, A. Messina, SafeSlice: a model slicing and design safety inspection tool for SysML, in: Proceedings of the 19th ACM SIGSOFT Symposium and the 13th European Conference on Foundations of Software Engineering (FSE), 2011, pp. 460–463.
[PS41] E. Denney, G. Pai, I. Habli, Towards measurement of confidence in safety cases, in: International Symposium on Empirical Software Engineering and Measurement (ESEM), 2011, pp. 380–383.
[PS42] E. Denney, S. Trac, A software safety certification tool for automatically generated guidance, navigation and control code, in: Aerospace Conference, 2008, IEEE, pp. 1–11.
[PS43] E. Denney, G. Pai, A lightweight methodology for safety case assembly, in: Computer Safety, Reliability, and Security, Springer, 2012, pp. 1–12.
[PS44] G. Despotou, M. Bennett, T. Kelly, Evaluation and Integration of COTS in Evidence based Assurance Frameworks, in: Making Systems Safer, Springer, 2010, pp. 233–254.
[PS45] A. Dick, S. Wills, Evidence-based development-applying safety engineering techniques to the progressive assurance and certification of complex systems, in: 3rd IET International Conference on System Safety, 2008, pp. 1–6.
[PS46] T. Dittel, H.-J. Aryus, How to "Survive" a safety case according to ISO 26262, in: Computer Safety, Reliability, and Security, Springer, 2010, pp. 97–111.
[PS47] I. Dodd, I. Habli, Safety certification of airborne software: an empirical study, Reliab. Eng. Syst. Safety 98 (2012) 7–23.
[PS48] K. Eastaughffe, A. Cant, M. Ozols, A framework for assessing standards for safety critical computer-based systems, in: 4th International Symposium and Forum on Software Engineering Standards, 1999, pp. 33–44.
[PS49] E. El Koursi, G. Mariano, Assessment and certification of safety critical software, in: Proceedings of the 5th Biannual World Automation Congress, 2002, pp. 51–57.
[PS50] E. El Koursi, P. Meganck, Assessment criteria for safety critical computer, in: Systems, Man, and Cybernetics, International Conference on. IEEE, 1998, pp. 3240–3245.
[PS51] L.-H. Eriksson, Using formal methods in a retrospective safety case, in: Computer Safety, Reliability, and Security, Springer, 2004, pp. 31–44.
[PS52] C. Esposito, D. Cotroneo, R. Barbosa, N. Silva, Qualification and selection of off-the-shelf components for safety critical systems: a systematic approach, in: 5th Latin-American Symposium on Dependable Computing Workshops (LADCW), 2011, pp. 52–57.
[PS53] J. Evans, T. Kelly, Defence standard 00–56 issue 4 and civil standards-appropriateness and sufficiency of evidence, in: The 3rd IET International Conference on System Safety 2008, pp. 43–43.
[PS54] D. Falessi, M. Sabetzadeh, L. Briand, E. Turella, T. Coq, R. Panesar-Walawege, Planning for safety evidence collection: a tool-supported approach based on modelling of standards compliance information, IEEE Softw. (2011) 99.
[PS55] M.S. Feather, L.Z. Markosian, Building a safety case for a safety-critical NASA space vehicle software system, in: 4th International Conference on Space Mission Challenges for Information Technology (SMC-IT), 2011, pp. 10–17.
[PS56] P.H. Feiler, Model-based validation of safety-critical embedded systems, in: Aerospace Conference, IEEE, 2010, pp. 1–10.
[PS57] J. Fenn, B. Jepson, Putting trust into safety arguments, in: Constituents of Modern System-safety Thinking, Springer, 2005, pp. 21–35.
[PS58] N. Fenton, B. Littlewood, M. Neil, L. Strigini, A. Sutcliffe, D. Wright, Assessing dependability of safety critical systems using diverse evidence, in: Software, IEE Proceedings-, IET, 1998, pp. 35–39.
[PS59] T. Ferrell, U. Ferrell, Use of service history for certification credit for COTS, in: 20th Conference on Digital Avionics Systems (DASC), vol. 1, 2001, pp. 1B1/1–1B1/7.
[PS60] M. Forster, M. Trapp, Fault tree analysis of software-controlled component systems based on second-order probabilities, in: 20th International Symposium on Software Reliability Engineering (ISSRE), 2009, pp. 146–154.
[PS61] D. Fowler, P. Bennett, IEC 61508—a suitable basis for the certification of safety-critical transport-infrastructure systems? in: Computer Safety, Reliability and Security, Springer, 2000, pp. 250–263.
[PS62] A. Galloway, R.F. Paige, N. Tudor, R. Weaver, I. Toyn, J. McDermid, Proof vs testing in the context of safety standards, in: 24th International conference on Digital Avionics Systems Conference (DASC), vol. 12, 2005, pp. 14.
[PS63] J. Good, A. Blandford, Incorporating human factors concerns into the design and safety engineering of complex control systems, in: International Conference on Human Interfaces in Control Rooms, Cockpits and Command Centres, 1999, pp. 51–56.
[PS64] P.J. Graydon, J.C. Knight, E.A. Strunk, Achieving dependable systems by synergistic development of architectures and assurance cases, in: Architecting Dependable Systems IV, Springer, 2007, pp. 362–382.
[PS65] P.J. Graydon, J.C. Knight, E.A. Strunk, Assurance based development of critical systems, in: 37th Annual International Conference on Dependable Systems and Networks (DSN), 2007, pp. 347–357.
[PS66] I. Habli, T. Kelly, A generic goal-based certification argument for the justification of formal analysis, Electron. Notes Theor. Comput. Sci. 238 (2009) 27–39.
[PS67] I. Habli, T. Kelly, A model-driven approach to assuring process reliability, in: 19th International Symposium on Software Reliability Engineering (ISSRE), 2008, pp. 7–16.
[PS68] I. Habli, T. Kelly, Achieving integrated process and product safety arguments, in: The Safety of Systems, Springer, 2007, pp. 55–68.
[PS69] I. Habli, T. Kelly, Process and product certification arguments: getting the balance right, ACM SIGBED Rev. 3 (2006) 1–8.
[PS70] I. Habli, W. Wu, K. Attwood, T. Kelly, Extending argumentation to goal-oriented requirements engineering, in: Advances in Conceptual Modeling-Foundations and Applications, Springer, 2007, pp. 306–316.

Please cite this article in press as: S. Nair et al., An extended systematic literature review on provision of evidence for safety certification, Inform. Softw. Technol. (2014), http://dx.doi.org/10.1016/j.infsof.2014.03.001

[PS71] J.G. Hall, L. Rapanotti, Assurance-driven design, in: the 3rd International Conference on Software Engineering Advances (ICSEA), 2008, pp. 379–388.

[PS72] V. Hamilton, Accounting for evidence: managing evidence for goal based software safety standards, in: Advances in Systems Safety, Springer, 2011, pp. 41–51.

[PS73] H. Harju, J. Lahtinen, J. Ranta, R. Nevalainen, M. Johansson, Software safety standards for the basis of certification in the nuclear domain, in: 7th International Conference on Quality of Information and Communications Technology (QUATIC), 2010, pp. 54–62.

[PS74] R. Hawkins, T. Kelly, J. Knight, P. Graydon, A new approach to creating clear safety arguments, in: Advances in Systems Safety, Springer, 2011, pp. 3–23.

[PS75] R. Hawkins, T. Kelly, A structured approach to selecting and justifying software safety evidence, in: 5th IET International Conference on System Safety 2010, pp. 1–6.

[PS76] R. Hawkins, T. Kelly, Software safety assurance-what is sufficient? in: 4th IET International Conference on Systems Safety, 2009, pp. 23–23.

[PS77] K.J. Hayhurst, D.S. Veerhusen, A practical approach to modified condition/decision coverage, in: 20th Conference on Digital Avionics Systems (DASC), vol. 11, 2001, pp. 1B2/1-1B2/10.

[PS78] M.P. Heimdahl, Safety and software intensive systems: Challenges old and new, in, Future of Software Engineering, IEEE Comput. Soc. 2007 (2007) 137–152.

[PS79] J. Hill, S. Tilley, Creating safety requirements traceability for assuring and recertifying legacy safety-critical systems, in: the 18th International Conference on Requirements Engineering (RE), 2010, pp. 297–302.

[PS80] C.M. Holloway, Safety case notations: alternatives for the non-graphically inclined? in: 3rd IET International Conference on System Safety, 2008, pp. 1–6.

[PS81] Z. Hu, C.G. Bilich, Experience with establishment of reusable and certifiable safety lifecycle model within abb, in: Computer Safety, Reliability, and Security, Springer, 2009, pp. 132–144.

[PS82] M. Huhn, A. Zechner, Analyzing dependability case arguments using quality models, Comput. Safety Reliab. Security Lect. Notes Comput. Sci. 5775 (2009) 118–131.

[PS83] M. Huhn, A. Zechner, Arguing for software quality in an IEC 62304 compliant development process, in: Leveraging Applications of Formal Methods, Verification, and Validation, Springer, 2010, pp. 296–311.

[PS84] E. Jee, I. Lee, O. Sokolsky, Assurance cases in model-driven development of the pacemaker software, in: Leveraging Applications of Formal Methods, Verification, and Validation, Springer, 2010, pp. 343–356.

[PS85] M. Johansson, R. Nevalainen, Additional requirements for process assessment in safety–critical software and systems domain, J. Softw.: Evolut. Process 24 (2012) 501–510.

[PS86] G. Jolliffe, Producing a safety case for IMA blueprints, in: The 24th International Conference on Digital Avionics Systems Conference (DASC), vol. 12, 2005, pp. 14.

[PS87] E. Joung, S. Oh, S. Park, G. Kim, Safety criteria and development methodology for the safety critical railway software, in: 31st International Conference on Telecommunications Energy Conference (INTELEC), 2009, pp. 1–4.

[PS88] D. Karydas, A. Brombacher, Reliability certification of programmable electronic systems, Reliab. Eng. Syst. Safety 66 (1999) 103–107.

[PS89] T.P. Kelly, Managing complex safety cases, in: Current Issues in Safety-Critical Systems, Springer, 2003, pp. 99–115.

[PS90] T. Kelly, Can process-based and product-based approaches to software safety certification be reconciled? in: Improvements in System Safety, Springer, 2008, pp. 3–12.

[PS91] E. Kesseler, Assessing COTS software in a certifiable safety-critical domain, Inform. Syst. J. 18 (2008) 299–324.

[PS92] S. Kinnersly, Safety Cases–what can we learn from Science? in: Advances in Systems Safety, Springer, 2011, pp. 25–40.

[PS93] A. Kornecki, J. Zalewski, Certification of software for real-time safety-critical systems: state of the art, Innovations Syst. Softw. Eng. 5 (2009) 149–161.

[PS94] G. Kotonya, I. Sommerville, Integrating safety analysis and requirements engineering, in: Proceedings on International Computer Science Conference (ICSC) and Asia Pacific Software Engineering Conference (APSEC), 1997, pp. 259–271.

[PS95] D. Kritzinger, Safety cases & safety assessments, in: 4th IET International Conference on Systems Safety, 2009.

[PS96] S. Kuball, G. Hughes, Decision-support for certification by calculating the evidential volume of a product, in: Proceedings on International Conference on Dependable Systems and Networks, 2003, pp. 15–24.

[PS97] J. Lahtinen, M. Johansson, J. Ranta, H. Harju, R. Nevalainen, Comparison between IEC 60880 and IEC 61508 for certification purposes in the nuclear domain, in: Computer Safety, Reliability, and Security, Springer, 2010, pp. 55–67.

[PS98] J.D. Lawrence, W.L. Persons, G.G. Preckshot, J. Gallagher, Evaluating software for safety systems in nuclear power plants, in: Proceedings of the Ninth Annual Conference on Computer Assurance Safety, Reliability, Fault Tolerance, Concurrency and Real Time, Security (COMPASS), 1994, pp. 197–207.

[PS99] R. Lewis, Safety case development as an information modelling problem, in: Safety-Critical Systems: Problems, Process and Practice, Springer, 2009, pp. 183–193.

[PS100] S. Linling, T. Kelly, Safety arguments in aircraft certification, in: 4th IET International Conference on Systems Safety, 2009, pp 31–31.

[PS101] B. Littlewood, D. Wright, 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. Softw. Eng. 33 (2007) 347–365.

[PS102] S. Liu, V. Stavridou, B. Dutertre, The practice of formal methods in safety-critical systems, J. Syst. Softw. 28 (1995) 77–87.

[PS103] J. Lucas, Safety case experiences from Harrier, in: Improvements in System Safety, Springer, 2008, pp. 77–91.

[PS104] R. Lutz, A. Patterson-Hine, Using fault modeling in safety cases, in: 19th International Symposium on: Software Reliability Engineering (ISSRE), 2008, pp. 271–276.

[PS105] D. Mannering, J.G. Hall, L. Rapanotti, Safety process improvement with POSE and Alloy, in: Improvements in System Safety, Springer, 2008, pp. 25–41.

[PS106] P. Mayo, Creating a competence argument to support a safety case, in: 4th IET International Conference on Systems Safety, 2009, pp. 23–23.

[PS107] J. McDermid, Proving the design in the safety case, in: Designing Safety-Critical Systems, IEE Colloquium on. IET, 1994, pp. 7/1–7/4.

[PS108] J. McDermid, Safety arguments, software and system reliability, in: The International Symposium on Software, Reliability Engineering, 1991, pp. 43–50.

[PS109] J.A. McDermid, Software safety: where's the evidence? in: Proceedings of the Sixth Australian Workshop on Safety Critical Systems and Software-Volume 3, Australian Computer Society Inc, 2001, pp. 1–6.

[PS110] S. McDonnell, B.E. Melhart, Software assessment to support certification for an existing computer-based system, in: Proceedings on International Symposium and Workshop on Engineering of Computer-Based Systems, 1996, pp. 190–197.

[PS111] D. Meacham, J. Michael, M.-T. Shing, J. Voas, Standards interoperability: applying software safety assurance standards to the evolution of legacy software, in: International Conference on System of Systems Engineering (SoSE), 2009, pp. 1–8.

[PS112] C. Menon, R. Hawkins, J. McDermid, Defence standard 00–56 issue 4: Towards evidence-based safety standards, in: Safety-Critical Systems: Problems, Process and Practice, Springer, 2009, pp. 223–243.

[PS113] C. Menon, J. McDermid, P. Hubbard, Goal-based safety standards and cots software selection, in: 4th IET International Conference on Systems Safety 2009. pp. 22–22.

[PS114] D. Méry, N.K. Singh, Trustable formal specification for software certification, in: Leveraging Applications of Formal Methods, Verification, and Validation, Springer, 2010, pp. 312–326.

[PS115] R. Moraes, J. Durães, E. Martins, H. Madeira, Component-based software certification based on experimental risk assessment, in: Dependable Computing, Springer, 2007, pp. 179–197.

[PS116] J.P. Near, A. Milicevic, E. Kang, D. Jackson, A lightweight code analysis and its role in evaluation of a dependability case, in: The 33rd International Conference on Software Engineering (ICSE), 2011, pp. 31–40.

[PS117] E.A. Nguyen, A.G. Ellis, Experiences with assurance cases for spacecraft safing, in: The 22nd International Symposium on Software Reliability Engineering (ISSRE), 2011, pp. 50–59.

[PS118] O. Nordland, Presenting a safety case—a case study, in: Computer Safety, Reliability and Security, Springer, 2001, pp. 56–65.

[PS119] A. Ogunsola, S. Pomeroy, EMC assurance and safety critical apparatus in a railway environment, in: International Symposium on Electromagnetic Compatibility (EMC), 2003, pp. 429–432.

[PS120] R. Palin, I. Habli, Assurance of automotive safety–a safety case approach, in: Computer Safety, Reliability, and Security, Springer, 2010, pp. 82–96.

[PS121] R.K. Panesar-Walawege, M. Sabetzadeh, L. Briand, T. Coq, Characterizing the chain of evidence for software safety cases: A conceptual model based on the IEC 61508 standard, in: The 3rd International Conference on Software Testing, Verification and Validation (ICST), 2010, pp. 335–344.

[PS122] R.K. Panesar-Walawege, M. Sabetzadeh, L. Briand, A model-driven engineering approach to support the verification of compliance to safety standards, in: The 22nd International Symposium on Software Reliability Engineering (ISSRE), 2011, pp. 30–39.

[PS123] Y. Papadopoulos, Model-based system monitoring and diagnosis of failures using statecharts and fault trees, Reliab. Eng. Syst. Safety 81 (2003) 325–341.

[PS124] Y. Papadopoulos, J.A. McDermid, The potential for a generic approach to certification of safety critical systems in the transportation sector, Reliab. Eng. Syst. Safety 63 (1999) 47–66.

[PS125] R. Pierce, H. Baret, Structuring a safety case for an air traffic control operations room, in: Constituents of Modern System-safety Thinking, Springer, 2005, pp. 51–64.

[PS126] C. Pygott, S.P. Wilson, Justifying reliability claims for a fault-detecting parallel architecture, J. Syst. Architect. 43 (1997) 735–751.

[PS127] F. Redmill, Analysis of the COTS debate, Saf. Sci. 42 (2004) 355–367.

[PS128] S.L.D. Reinhardt, J. McDermid, Assurance of claims and evidence for aviation systems, in: the 5th IET International Conference on System Safety, 2010, pp. 1–10.

[PS129] D. Reinhardt, Certification criteria for emulation technology in the australian defence force military avionics context, in: Proceedings of the

Eleventh Australian Workshop on Safety Critical Systems and Software-Volume 69, Australian Computer Society Inc, 2007, pp. 79–92.

[PS130] K. Rich, H. Blanchard, J. McCloskey, The use of goal structuring notation as a method for ensuring that human factors is represented in a safety case, in: 2nd IET International Conference on system Safety, 2007, pp. 217–222.

[PS131] W. Ridderhof, H.G. Gross, H. Doerr, Establishing evidence for safety cases in automotive systems – a case study, in: Computer Safety, Reliability, and Security, Springer, 2007, pp. 1–13.

[PS132] L.K. Rierson, Object-oriented technology (OOT) in civil aviation projects: certification concerns, in: Proceedings of the 18th International conference on Digital Avionics Systems Conference, vol. 1, 1999, pp. 2. C. 4-1-2. C. 4–8.

[PS133] J. Rushby, Formalism in safety cases, in: Making Systems Safer, Springer, 2010, pp. 3–17.

[PS134] J. Rushby, Just-in-time certification, in: 12th IEEE International Conference on Engineering Complex Computer Systems, 2007, pp. 15–24.

[PS135] J. Rushby, New challenges in certification for aircraft software, in: Proceedings of the 9th ACM international conference on Embedded software, ACM, 2011, pp. 211–218.

[PS136] J. Rushby, Runtime certification, in: Runtime Verification, Springer, 2008, pp. 21–35.

[PS137] M. Sabetzadeh, D. Falessi, L. Briand, S.D. Alesio, D. McGeorge, V. Ahjem, J. Borg, Combining goal models, expert elicitation, and probabilistic simulation for qualification of new technology, in: The 13th International Symposium on High-Assurance Systems Engineering (HASE), 2011, pp. 63–72.

[PS138] A. Saeed, R. de Lemos, T. Anderson, On the safety analysis of requirements specifications for safety-critical software, ISA Trans. 34 (1995) 283–295.

[PS139] D. Schneider, M. Trapp, A safety engineering framework for open adaptive systems, in: The 5th International Conference on Self-Adaptive and Self-Organizing Systems (SASO), 2011, pp. 89–98.

[PS140] E. Schoitsch, E. Althammer, H. Eriksson, J. Vinter, L. Gönczy, A. Pataricza, G. Csertan, Validation and Certification of Safety-Critical Embedded Systems-The DECOS Test Bench, in: Computer Safety, Reliability, and Security, Springer, 2006, pp. 372–385.

[PS141] R. Shaw, Safety-critical software and current standards initiatives, Comput. Methods Programs Biomed. 44 (1994) 5–22.

[PS142] M. Sherriff, L. Williams, Devcop: a software certificate management system for eclipse, in: The 17th International Symposium on Software Reliability Engineering (ISSRE), 2006, pp. 375–384.

[PS143] S.P. Smith, M.D. Harrison, B.A. Schupp, How explicit are the barriers to failure in safety arguments? in: Computer Safety, Reliability, and Security, Springer, 2004, pp. 325–337.

[PS144] J. Spriggs, Developing a safety case for autonomous vehicle operation on an airport, in: Current Issues in Safety-Critical Systems, Springer, 2003, pp. 79–98.

[PS145] M.J. Squair, Issues in the application of software safety standards, in: Proceedings of the 10th Australian Workshop on Safety Critical Systems and Software-Volume 55, Australian Computer Society Inc, 2006, pp. 13–26.

[PS146] H. Stallbaum, M. Rzepka, Toward DO-178B-compliant Test Models, in: The International Workshop on Model-Driven Engineering, Verification, and Validation (MoDeVVa), 2010, pp. 25–30.

[PS147] E. Stensrud, T. Skramstad, J. Li, J. Xie, Towards goal-based software safety certification based on prescriptive standards, in: The 1st International Workshop on Software Certification (WoSoCER), 2011, pp. 13–18.

[PS148] Z. Stephenson, J. McDermid, Supporting explicit interpretation of standards and guidance, in: 5th IET International Conference on System Safety, 2010. pp. 33–33.

[PS149] Z. Stephenson, C. Fairburn, G. Despotou, T. Kelly, N. Herbert, B. Daughtrey, Distinguishing fact from fiction in a system of systems safety case, in: Advances in Systems Safety, Springer, 2011, pp. 55–72.

[PS150] M.-A. Sujan, F. Koornneef, U. Voges, Goal-based safety cases for medical devices: opportunities and challenges, in: Computer Safety, Reliability, and Security, Springer, 2007, pp. 14–27.

[PS151] S. Linling, Z. Wenjin, T. Kelly, Do safety cases have a role in aircraft certification?, Proc Eng. 17 (2011) 358–368.

[PS152] T. Cockram, B. Lockwood, Electronic safety cases: challenges and opportunities, in: Current Issues in Safety-Critical Systems, Springer, 2003, pp. 151–162.

[PS153] T. Kelly, J. McDermid, A systematic approach to safety case maintenance, Reliab. Eng. Syst. Safety 71 (2001) 271–284.

[PS154] T. Yuan, T. Xu, Computer system safety argument schemes, in: the Second World Congress on Software Engineering (WCSE), 2010, pp. 107–110.

[PS155] T. Kelly, Using software architecture techniques to support the modular certification of safety-critical systems, in: Proceedings of the 11th Australian Workshop on Safety critical Systems and Software-Volume 69, Australian Computer Society Inc, 2007, pp. 53–65.

[PS156] F. Torner, P. Ohman, Automotive safety case a qualitative case study of drivers, usages, and issues, in: The 11th International High Assurance Systems Engineering Symposium (HASE), pp. 313–322.

[PS157] J. L. Valk, H. Vis, G. Koning, Phileas, A safety critical trip around the world, in: Making Systems Safer, Springer, 2010, pp. 115–126.

[PS158] P. Varley, Techniques for development of safety-related software for surgical robots, Inform. Technol. Biomed., IEEE Trans. 3 (1999) 261–267.

[PS159] S. Wagner, B. Schatz, S. Puchner, P. Kock, A case study on safety cases in the automotive domain: Modules, patterns, and models, in: the 21st International Symposium on Software Reliability Engineering (ISSRE), 2010, pp. 269–278.

[PS160] J. Wang, Offshore safety case approach and formal safety assessment of ships, J. Safety Res. 33 (2002) 81–115.

[PS161] R. Weaver, G. Despotou, T. Kelly, J. McDermid, Combining software evidence: arguments and assurance, in: ACM SIGSOFT Software Engineering Notes, ACM, 2005, pp. 1–7.

[PS162] R. Weaver, J. Fenn, T. Kelly, A pragmatic approach to reasoning about the assurance of safety arguments, in: Proceedings of the 8th Australian workshop on Safety critical systems and software-Volume 33, Australian Computer Society Inc, 2003, pp. 57–67.

[PS163] R. Weaver, T. Kelly, P. Mayo, Gaining confidence in goal-based safety cases, in: Developments in Risk-based Approaches to Safety, Springer, 2006, pp. 277–290.

[PS164] S. Wilson, J.A. McDermid, P. Kirkham, P. Fenelon, The Safety Argument Manager: an integrated approach to the engineering and safety assessment of computer based systems, in: Proceedings of the International Symposium and Workshop on Engineering of Computer-Based Systems, 1996, pp. 198–205.

[PS165] W. Winkelbauer, G. Schedl, A. Gerstinger, Safety case practice-meet the challenge, in: Developments in Risk-based Approaches to Safety, Springer, 2006, pp. 83–104.

[PS166] J. Wlad, Software reuse in safety-critical airborne systems, in: The 25th Digital Avionics Systems Conference, IEEE/AIAA, 2006, pp. 1–8.

[PS167] W. Wu, T. Kelly, Combining Bayesian Belief Networks and the goal structuring notation to support architectural reasoning about safety, in: Computer Safety, Reliability, and Security, Springer, 2007, pp. 172–186.

[PS168] W. Wu, T. Kelly, Towards evidence-based architectural design for safety-critical software applications, in: Architecting Dependable Systems IV, Springer, 2007, pp. 383–408.

[PS169] F. Yan, Comparison of means of compliance for onboard software certification, in: The 4th International Conference on Computer Science & Education (ICCSE), 2009, pp. 917–920.

[PS170] F. Ye, T. Kelly, Contract-based justification for COTS component within safety-critical applications, in: Proceedings of the 9th Australian workshop on Safety Critical Systems and Software-Volume 47, Australian Computer Society Inc, 2004, pp. 13–22.

[PS171] S. Yih, C.F. Fan, Analysing the decision making process of certifying digital control systems of nuclear power plants, Nucl. Eng. Des. 242 (2012) 379–388.

[PS172] G. Zoughbi, L. Briand, Y. Labiche, Modelling safety and airworthiness (RTCA DO-178B) information: conceptual model and UML profile, Softw. Syst. Modell. 10 (2011) 337–367.

[PS173] R. Palin, D. Ward, I. Habli, R. Rivett, ISO 26262 safety cases: compliance and assurance, in: the 6th Internal Conference on System Safety, 2011, 1–6.

[PS174] P. Graydon, I. Habli, R. Hawkins, T. Kelly, J. Knight, Arguing conformance, IEEE Softw. 29 (2012) 50–57.

[PS175] E. Denney, G. Pai, I. Habli, Perspectives on software safety case development for unmanned aircraft, in: The 42nd Annual International Conference on Dependable Systems and Networks (DSN), 2012, pp. 1–8.

[PS176] L. Cyra, J. Górski, Supporting compliance with security standards by trust case templates, in: The 2nd International Conference on Dependability of Computer Systems (DepCoS-RELCOMEX), 2007, pp. 91–98.

[PS177] C. Hobbs, M. Lloyd, The application of Bayesian Belief Networks to assurance case preparation, in: Achieving Systems Safety, Springer, 2012, pp. 159–176.

[PS178] M. Thomas, Unsafe standardization, Computer 40 (2007) 109–111.

[PS179] M. Åkerholm, R. Land, Towards systematic software reuse in certifiable safety-critical systems, in: International Workshop on Software Reuse and Safety (RESAFE), Falls Church, VA, 2009.

[PS180] I. Bate, S. Bates, J. McDermid, Safety arguments for use of an Ada to FPGA compiler, in: Proceedings of the 22nd International System Safety Conference, 2004. pp. 685–694.

[PS181] I. Bate, P. Conmy, T. Kelly, J. McDermid, Use of modern processors in safety-critical applications, Comput. J. 44 (2001) 531–543.

[PS182] M. Beine, A Model-based reference workflow for the development of safety-critical software, in Embedded Real Time Software and Systems, 2010. pp. 1–6.

[PS183] P. Bishop, R. Bloomfield, S. Guerra, The future of goal-based assurance cases, in: Proceedings Workshop on Assurance Cases, 2004, pp. 390–395.

[PS184] N. Limnios, Maintenance optimisation of A digital engine control system with limit failure rate constrain, in: 22nd Congress of International Council of the Aeronautical Sciences, Harrogate, 2000.

[PS185] R. Brown, Improving the production and presentation of safety cases through the use of intranet technology, in: Industrial Perspectives of Safety-critical Systems, Springer, 1998, pp. 184–193.

[PS186] D. Bush, A. Finkelstein, Reuse of safety case claims-an initial investigation, in: Proceedings of the London Communications Symposium. University College London, 2001.

[PS187] N. Cameron, M. Webster, M. Jump, M. Fisher, Certification of a civil UAS: a virtual engineering approach, in: Proceedings of the 2011 AIAA Modelling SImulation and Technologies Conference and Exhibit, AIAA, Portland, Oregon, 2011, pp. 1–15.

[PS188] G. Cleland, J. Blanquart, J. Carranza, P. Froome, C. Jones, J. Muller, A framework for the software aspects of the safety certification of a space system, in: Joint ESA-NASA Space-Flight Safety Conference, 2002, pp. 175.

[PS189] D.J. Coe, J.S. Hogue, J.H. Kulick, Software Safety Engineering Education, world-comp.org, 2011. <http://world-comp.org/p2011/SER4081.pdf>.

[PS190] J. Dehlinger, R.R. Lutz, Bi-directional safety analysis for product-line, multi-agent systems, in: ACM SIGBED Review: Special Issues on Workshop Innovative Techniques for Certification of Embedded Systems, 2006.

[PS191] I. Fey, J. Mller, M. Conrad, Model-based design for safety-related applications, in: Proceedings of SAE Convergence, 2008.

[PS192] A. Shen, K. Makarychev, Y.S. Makarychev, The importance of being formal, Math. Intell. 23 (2001) 41–42.

[PS193] W. S. Greenwell, A taxonomy of fallacies in system safety arguments, in: Proceedings of the 2006 International System Safety Conference, 2006.

[PS194] C. Gurr, Argument representation for dependable computer-based systems, Informal Logic (2001) 22.

[PS195] I. Habli, I. Ibarra, R. Rivett, T. Kelly, Model-based assurance for justifying automotive functional safety, in: Proceedings 2010 SAE World Congress, 2010.

[PS196] I. Habli, R. Hawkins, T. Kelly, Software safety: relating software assurance and software integrity, Int. J. Crit. Comput.-Based Syst. 1 (2010) 364–383.

[PS197] P. Hollow, J. McDermid, M. Nicholson, Approaches to certification of reconfigurable IMA systems, in: Proceedings 10th International Symposium of the International Council on Systems Engineering, 2000.

[PS198] S.A. Jacklin, Closing the certification gaps in adaptive flight control software, in: Proceedings AIAA Guidance, Navigation and Control Conf. 2008.

[PS199] O. Kath, R. Schreiner, J. Favaro, Safety, security, and software reuse: a model-based approach, in: 4th International Workshop in Software Reuse and Safety (RESAFE), Washington, DC, USA, 2009.

[PS200] V. Katta, T. Stalhane, A conceptual model of traceability for safety systems, in: CSDM-Poster Presentation, 2010.

[PS201] T. Kelly, A systematic approach to safety case management, in: Proceedings of SAE 2004 World Congress, Detroit, MI, Citeseer, 2004.

[PS202] T. Kelly, R. Weaver, The goal structuring notation–a safety argument notation, in: Proceedings of the Dependable Systems and Networks 2004 Workshop on Assurance Cases, Citeseer, 2004.

[PS203] O. Lisagor, J. McDermid, D. Pumfrey, Towards a practicable process for automated safety analysis, in: 24th International System Safety Conference, Citeseer, 2006, pp. 596–607.

[PS204] M. Nicholson, J. McDermid, Extending PSSA for complex systems, in: Proceedings of the 21st International System Safety Conference (ISSC), 2003.

[PS205] M. Nicholson, P. Conmy, I. Bate, J. McDermid, Generating and maintaining a safety argument for integrated modular systems, in: 5th Australian Workshop on Industrial Experience with Safety Critical Systems and Software, Melbourne, Australia, 2000, pp. 31–41.

[PS206] J. Bohn, W. Damm, J. Klose, A. Moik, H. Wittke, Modeling and validating train system applications using statemate and live sequence charts, in: Proceedings IDPT, Citeseer, 2002.

[PS207] M. Ozols, K. Eastaughffe, A. Cant, S. Collignon, DOVE: a tool for design modelling and verification in safety critical systems, in: 16th International System Safety Conference, Citeseer, 1998.

[PS208] J. Brunel, C. Jacques, Formal verification of a safety argumentation and application to a complex UAV system, in: Computer Safety, Reliability, and Security, Springer, Berlin Heidelberg, 2012, pp. 307–318.

[PS209] J. Rushby, how do we certify for the unexpected? in: AIAA Guidance, Navigation and Control Conference and Exhibit, 2008.

[PS210] Z. Stephenson, T. Kelly, J. Camus, Developing an argument for Def Stan 00–56 from existing qualification evidence, in: Embedded Real-Time Software and Systems, 2010.

[PS211] S.A. Vilkomir, V.S. Kharchenko, An 'Asymmetric'approach to the assessment of safety-critical software during certification and licensing, in: Project Control: the Human Factor, Proceedings of ESCOM–SCOPE Conference, 2000, pp. 18–20.

[PS212] J. Wang, Analysis of safety-critical software elements in offshore safety studies, Disaster Prevent. Manag. 9 (2000) 271–282.

[PS213] L. Whiting, M. Hill, Safety analysis of hawk in flight monitor, in: ACM SIGSOFT Software Engineering Notes, ACM, 1999, pp. 32–38.

[PS214] L. Wildman, T. Cant, C. Edwards, A. Griffiths, B. Mahony, B. Martin, A. Rae, Guidance for Def (Aust) 5679 Issue 2, in: 13th Australian Conference on Safety Related Programmable Systems, Australian Computer Society, System Safety and Quality Engineering Pty Ltd, 2008.

[PS215] S. Wilson, T.P. Kelly, J.A. McDermid, Safety case development: Current practice, future prospects, in: Safety and Reliability of Software Based Systems, Springer, 1997, pp. 135–156.

[PS216] F. Ye, T. Kelly, Use of COTS Software Components in Safety-critical Applications-a Defensible Approach, 2004.

[PS217] T. Yuan, T. Kelly, Argument schemes in computer system safety engineering, Informal Logic 31 (2011) 89–109.

[PS218] R. Weaver, J. McDermid, T. Kelly, Software safety arguments: towards a systematic categorisation of evidence, in: the 20th International System Safety Conference, Denver, CO, Citeseer, 2002.

Please cite this article in press as: S. Nair et al., An extended systematic literature review on provision of evidence for safety certification, Inform. Softw. Technol. (2014), http://dx.doi.org/10.1016/j.infsof.2014.03.001