ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 9, September 2015

A Literature Review on Assurance Driven Software Design

Dipak Gade[1], Dr. Santosh Deshpande[2]

Department of CSE, Shri Jadishprasad Tibrewala University, Chudela, India [1]

Department of CA, MES IMCC, Savitribai Phule Pune University, Pune, India [2]

Abstract: Modern critical software systems are more complex and needs to fulfil constantly increasing demands from its users. This is forcing the system architects and software developers to design more reliable and trustworthy software systems which can stand long enough to meet expectations from all of its stakeholders. Designing of such reliable and promising software is not easy, specifically considering the complexity, huge efforts and time required to develop it. Developers have now seriously realised the importance of designing robust and reliable software. From last some years, more research is going on to identify ways and techniques to design the trustworthy and reliable software which can be guaranteed fit for use under given operating constraints. The present paper has reviewed some of the research papers explaining prominent techniques and methodologies for assurance driven software design.

Keywords: Assurance Arguments, Claims & Evidence, Design Assurance, Goal Structure Notations, Safety Cases.

I. INTRODUCTION

For any critical Software System, software is the backbone for its proper functioning and thus the guarantee of successful operation of the system. This requires ensuring that the software must be operational throughout its lifecycle without failure or without degrading its performance and operational efficiency. Development of such software can be enabled by offering goal oriented software assurance. This can be achieved by using assurance driven software designing which is an approach to the systematic design and development for the software systems and their assurance arguments. To ensure reliable functioning of the software, developers of such software systems, thoroughly carry out risks assessment, hazards analysis, collection and creation of evidences, penetration testing, quality inspections, reliability checks, peer review, detailed documentation etc. By using Assurance Driven Design (ADD) approach, however, this goal can be very well achieved by developing assurance cases and arguments as a support evidence to conclude that the software system is absolutely fit for its intended use under the required operating context. Following ADD approach one can ensure taking into account the goals of all relevant stakeholders apart from the goals related to basic parameters such as system safety, security requirements and complete system functionality. This in a way makes it possible to ensure that the software system developed is fit for use in a complete sense.

In ADD the basic driving force is building comprehensive assurance cases for all the functionalities of the system. The Assurance case consists of following three basic elements

- An assurance goal or claim: This is the intended operation or ultimate achievement needed by the system under given conditions. For a good system design, it is required to identify all the goals which are required to be fulfilled by the system
- Evidence: This is the supporting evidence to show that the goal is satisfied. Results of analysis, development artefact, observation, demonstration, testing, simulation etc. can be used as evidence to prove that goal is achieved.
- An argument: This facilitates linking the evidence to the goal.

The assurance cases then are applied repeatedly to produce a hierarchic kind of structure with the overall goal at the root level for the real system. Evidence at one level becomes goal at the subsequent lower level which facilitates argument to be manageable at each level.

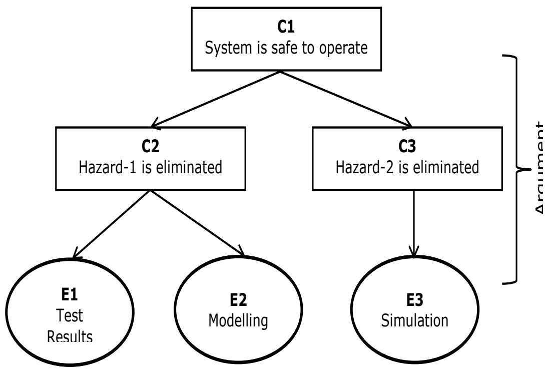

Fig. 1 Typical structure of an Assurance Case

The comprehensive assurance cases can also have additional elements such as appropriate strategies, justifications, assumptions made and the relevant context. Goal Structuring Notation (GSN) have been designed to facilitate writing assurance cases in a manner that is easy for humans to understand and that can be manipulated by machine. The typical structure of an assurance case with GSN is shown in Figure 1 above. In the Figure 1, the assurance case has represented that the high level goal of

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 9, September 2015

the system shown by Claim C1 that the system is safe to operate is based on the sub-claims C2 and C3 which argues that both the hazards 1 and 2 have been eliminated. The sub-claims C2 and C3 are based on the facts derived from evidences E1, E2 and E3 respectively. The elimination of hazard-1 is supported by evidence E1 and E2 whereas elimination of Hazard-2 is supported by evidence E3. The evidences can be in the form of test results, analysis, accepted prototype, modelling, simulation, V&V report etc.

II. LITERATURE REVIEW

This section has provided review of selected journal and conference papers available on software design using assurance driven approach and other relevant design assurance techniques.

[1] Peter Bishop, Robin Bloomfield and Sofia Guerra in their paper have discussed goal based approach and Adelard's approach for constructing safety cases and assurance cases. Authors have pointed out that in construction of safety/ assurance cases the primary focus should be the case contents and not just using of suitable notation to represent them. In Adelard approach for representing safety and assurance cases, the basic elements of claims, evidences and arguments can be used; also it is possible to have different kind of arguments such as deterministic, probabilistic quantitative statistical reasoning and qualitative compliance. Authors preferred using deterministic arguments and also highlighted the importance of separating arguments. It is pointed out that, for safety case, safety justification can have two types of claims that are claims about the system and claims about the safety case itself. Authors have proposed to separate the claims in two claim trees i.e. top level claim about system behaviour is safe and a tree concerning the quality of the safety case. The structure of the safety justifications can also be shaped by the modular assurance of system components, specifically off the shelf components. For all such components authors have recommended identifying of potential hazards during overall system hazard analysis and implementing suitable mitigations. Validation of the safety/ assurance case can be done considering use of formality and models. Authors have also provided possible future directions to improve frameworks for goal-based assurance cases.

[2] Scott and Krombolz have highlighted that for any safety/mission/security-critical systems, regulations and acquisition guidelines are available which demands documentary evidence that the system is safe to operate and shall satisfy the critical requirements without any issue. To produce documentary evidence about assurance and safety cases for the given system, authors selected a software notation suitable for building structured safety cases and they applied it to three different assurance standards. The paper has discussed each of the three standard mapping efforts along with the problems which were encountered by authors while developing the assurance and safety cases. In addition to the standards, authors also used the notation to structure an assurance case for a practical security-critical system. The paper has also described the lessons learned from this project. Authors have finally concluded with practical options for using the mappings of the standards and how well their initial hypotheses were borne out by the project.

[3] Emmet and Guerra have stated that for documenting critical requirements for the industrial systems, a documented case is require which is normally called as assurance case. Traditionally to certify the critical system against Safety certification normally a standard based compliance procedure was accepted practice and was considered as adequately safe to follow however this approach was mostly suitable for a stable environment where best practices are followed by extensive experience. For the industrial sectors which face fast moving technologies, authors have recommended to use goal based approach to develop the assurance case. Goal-based approaches are more flexible as they focus directly on the critical requirements and they are more attuned to the ways in which sophisticated engineering arguments are actually made. Assurance and safety cases are mostly matured concepts and are already have wide use to support the assurance of dependable systems. Authors have suggested that assurance case concepts are compatible with software certification process in that claims need to be established and backed up by evidence about the product or its development process. Authors have introduced a new tool which is specifically used to develop and manage assurance and safety cases. The tool is named as, ASCE (The Assurance and Safety Case Environment). The authors have clearly specified how ASCE could be used to support the development and management of software certificates.

[4] Graydon, Knight and Strunk in their paper have stated an Assurance Based development (ABD) methodology for the systematic designing of a critical computing system. An assurance case clearly specifies the dependability goals for the system and clarifies that the arguments links the available evidence justifying those system goals. If the system development goes in concurrence with the development of relevant assurance cases then it can greatly assist the developers to assess the technology options available which can address the specified dependable and critical goals of each component. It is pointed out by the authors that ABD can assure developers about the selection of appropriate technology which will support the system's critical goals and it can also offer the required flexibility to deploy the chosen technology for implementing the selected required components only which are critical and essential to meetassurance needs of the system. ABD simplifies meeting system dependability goals as system design progresses, rather than avoiding or addressing it after system development is complete as in case of traditional development way.

[5] Nguyen, Greenwell and Hecht have discussed use of Assurance case based development to address issues pertaining to transitioning from a legacy system to its alternate system will not compromise mission critical

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 9, September 2015

objectives of system. The application which was requiring transition was from the Global Positioning System (GPS) to a new AEP system which is basically a ground-control system. This transitioning imposes a challenge of ensuring continuous control of the GPS satellite constellation when the control was changed from the traditional mainframe system to a distributed architecture system. Authors solved this issue by developing an assurance case to restructure procedure based documentation into an easy to manage and analyse kind of documentation. The analysis concluded that the transition has not faced any critical hazards and this conclusion was fully validated by a successful completion of transition without any major issues. The authors found that risks identification using the safety cases was very effective and easy as compare to using legacy procedure based documentation to do the same. The system states used in the assurance case provided a direct subject. Authors have also stated that the assurance case enabled them to present and create their argument in a more systematic fashion; the act of creating the argument forced them to be more systematic in their thought procedure. Authors felt that representing Assurance Case with GSN was time consuming and bit difficult for understanding the argument structure.

[6] Hall and Rapanotti have discussed the Assurance Driven Design approach in detail. The authors have pointed out that intraditional design approach, developers first design the software and then tries to prove it through some evidence. Though this approach works some times, mostly it ends up in costly rework when the evidence is not strong enough to prove the develop software is trustworthy. In some cases there is a possibility of over-engineering of the systems which can increase the development cost. To avoid such issues authors have recommended adopting proactive design approach which is based on the assurance case. Assurance-driven design concludes assurance as a backbone behind the design process. Assurance-driven design guides developers on the systematic design which can guarantee system assurance. Authors made it clear that Assurance Driven Design approach is not descriptive process like approach rather it facilitates organisations to analyse their assurance needs as per their development requirement, including their risk taking, and their process adapting ability. The authors have clearly explained how problem and solutions validation can be used to manage the development risks and how assurance arguments can be implemented concurrently alongside while system development.

[7] Gandhi and Lee in their paper have pointed out an assurance case approach to document and present evidence during case study research design. Authors have also highlighted the fact that case studies have the potential to bridge between constructive and empirical approaches to requirement engineering research. In their paper, authors discussed the steps involved in case study research design based on the assurance case approach. The assurance notation was used to outline systematic way for planning the validation effort for a Requirements Engineering Methodology (REM). Authors have pointed out that the ideas contributed through their paper are for identifying the strengths of assurance cases while carrying out REM invention.

[8] Jee and Sokolsky in their paper have discussed the building up of an assurance case for the design of pacemaker software. The pacemaker software was developed using a model-based technique which combined formal system modelling, with code generation from the formal system model, and timing measurement of behaviour of the implementation. It is clarified that how the structure of the assurance case reacts the development approach. The author's presentedthe approach for the construction of assurance cases for the model-driven development of safety-critical software. The assurance case ties together all the evidence collected during the development process. Also for each stage of development that is model building, model code generation and model validation, a separate claim or set of claims were incorporated.

[9] Patrick J. Graydon and John C. Knight in their paper have described the Assurance Based Development (ABD) approach for constructing critical software systems and their assurance arguments.Authors have described in detail use of fitness argument of the system assurance case to prove that the system is acceptable and has all the safety and security relevant properties. In their paper, authors introduced the process synthesis mechanism of ABD with the success argument. The success argument documents the success criteria to verify that the system goal is acceptable without any issue. Authors have claimed that the success argument will safeguard developers from development risk.The ABD approach was demonstrated using the case study with design implementation of LifeFlow Left Ventricular Assist Device (LVAD). LifeFlow is an artificial heart pump designed for the longterm treatment of heart failure. Authors have stated that the implementation of LVAD control software using ABD approach has proved that ABD approach is feasible. Authors have also clarified that the unsupported goals in Assurance Arguments assisted them to determine if these are appropriate drivers for development choices whereas the ABD decision criteria's while implementing the LVAD case study assisted them determine if these are right criteria in evaluating different development choices.Authors concluded that ABD provides a comprehensive basis for development choices.

[10] Stringfellow, Leveson and Owens, have described the new hazard analysis technique called asSystems Theoretic Process Analysis (STPA) for development of software intensive systems. Authors have highlighted that traditionally, after system design, safety features are added in an attempt to prove that the system is reliable and safe for use. However, Safety is prime important and need to be designed from the beginning into a system; it cannot be added on to a mature design effectively. In addition, the intensive use of software is changing the way systems respond to the safety relevant issues and that makes it necessary to change the safety engineering techniques accordingly. The new hazard analysis technique - STPA,

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 9, September 2015

discussed in the paper, is suggested to be effective on software-intensive systems. Authors have pointed out that STPA has an advantage of driving the design decisions at early stage and can facilitate driving design decisions in parallel with design refinement. This turns out more effective approach in system design as compare to traditional system design process. Authors also felt that STPA is much economical as against conventional system design approach.

[11] José Luis Vivas, Isaac Agudo and Javier Lopez have introduced an assurance methodology that is suggested to integrate creating of assurance case along with development of system. Authors have claimed that this new methodology was developed in order to provide identity management, trust and assurance of privacy to the evolving European project Privacy and Identity Management for Community Services (PICOS). This methodology was developed by authors with an aim to develop an approach for maintaining and building system security aspects throughout its development life cycle in a typical system development life cycle. This also ensures producing relevant information for building up system assurance and feedback for system developers. The authors have clarified that this methodology is not a just a theoretical implementation and they did not face any practical or theoretical issues while developing the PICOS system. Authors have also pointed out that this new methodology will require further fine tuning and maturity while developing real world practical applications in future.

[12] P. Bieber, R. Delmas, and C. Seguinin their paper have described the Development Assurance Level allocation process (DAL). DAL indicates the level of rigorous process and details required for development of the software or hardware function for an Aircraft. It is associated with the item as per the classification of the most severe failure condition which can be caused by that item. The authors have proposed formalizing theory for the DAL allocation rules as per ARP475a recommended practices. There they have recommended DAL allocation based on two constraint satisfaction problems. These problems are mainly a) Identifying a minimal set of necessary independence relations between items and b) DAL allocation considering downgrading options due to the item independence relations identified by the first problem. Authors have claimed that the above stated CSPs can be efficiently solved by solvers by using java library, Sat4j, which is used for solving Boolean satisfaction and optimization problems. To clarify the concept authors modelled the DAL allocation problem using pseudo-Boolean logic. They introduced the predicate dalOk(mc, s) to represent that the chosen DAL allocation constraints can be satisfied by the subset s of mc. Authors explained the developed tool called DALculator which can be used for solving the two constraint satisfaction problems of independence identification and DAL allocation. Authors tested the DALculator and validated its results with the two class examples such as Data Measurement and Display system. Authors tested the DALculator and validated its results with the two class examples such as Data Measurement and Display system. They also tested the performance of DALculator using more complex systems which includes minimal cut sets computed for a Flight Control System and Electrical Distribution and Generation system.

[13] Wrona, Oudkerk, Hein, Menz and Ritter in their paper have described the development process for the High Assurance Attribute Based Access Control Guard (HAAG), which is one of the important security enablers in the NATO future information sharing architectures, including Information Exchange Gateway Scenario and Future Mission Networks. The HAAG implements Attribute-based Access Control (ABAC) for information requests, and enforces content-based protection and release policies. The system design process incorporates a structured way of collecting requirements and takes into account a security risk assessment of the system. The process is based on industry standards and best practices. It is accompanied by a definition of a Common Criteria Protection Profile, which captures security requirements for the HAAG. Authors have claimed that all phases of the system design process are performed using an integrated modelling environment based on Eclipse and open-source tools. It is also clarified by authors that the environment allows them to build and maintain a relatively complex model and, to a large extent, automatically generate the required design documentation.

[14] Saruwatari and Yamamoto have discussed about the D*Framework which is a method that can make assurance case for open system. The idea of D*Framework is based on the fact that the information systems which are developed as open system depend on each other and under such scenario Assurance cases are expected to confirm a dependability of open systems. To explain the applications and results and findings using D*Framework, authors had taken an example of an elevator system. An assurance case was developed for elevator system with two assurance cases of actors. The dependability information was obtained using D*Framework as actors, goals, strategies, solutions and contexts. Finally authors have shared that in the proposed example study they obtained “assured average” of 0.4 and “assured variance” of 0.84. In order to support creation of dependability case, the authors developed D-Case editor. In D-Case editor, an assurance case was created using Goal Structuring Notation.

[15] Devesh Bhatt, Gabor Madl, David Oglesby, Sam Owre, Natarajan Shankar and Ashish Tiwari discussed in their paper an assurance directed design approach by developing assurance cases for design of cyber physical system such as transportation system. Authors have insisted that development of assurance argument must be part of design to ensure early identification and resolving of risks. Authors have proposed a layered assurance arguments development where each abstraction layer can represent explicit assumptions and approximations made while system design. The abstraction layers can start with highest level representing top level system and second level representing models which captures engineering

approximations and inaccuracies. With further drill down, one can have third abstraction layer representing computational elements involving dataflow and temporal dependencies between computation and physical components whereas fourth layer represents mapping between computations to physical platform. Authors felt that these structured assurance cases as per the abstraction layers can assist in isolating the concerns and can allow global view of the system which can be further decomposed as per individual control and functionality. Authors have also introduced an Evidential Tool Bus (ETB), a tool integration platform aim at curation of evidences and from multiple synthesis tools and analysis claims. ETB can be applied to develop assurance workflows, integrated new tools and apply the workflows to prove the claims that are supported by evidence and arguments.

TABLE I SUMMARY OF LITERATURE REVIEW

Paper	Summary
[1]	Goal based approach and Adelard's approach for constructing safety cases and assurance cases.
[2]	Building of structured assurance and safety cases by using suitable software notation in order to produce documentary evidence.
[3]	Assurance case concepts are compatible with software certification process for the product or its development process.
[4]	Assurance Based development (ABD) methodology for the systematic designing of a critical computing system.
[5]	Risks identification using the safety cases was very effective and easy as compare to using legacy procedure based documentation to do the same.
[6]	Assurance-driven design guides developers on the systematic design which can guarantee system assurance.
[7]	Assurance case approach was very effective and proved as systematic way for planning the validation effort for a Requirements Engineering Methodology case study research design.
[8]	Effective utilisation of Assurance case approach for model driven development for designing of pacemaker software.
[9]	Process synthesis mechanism of Assurance Based Development with the success argument
[10]	The new hazard analysis technique - Systems Theoretic Process Analysis (STPA) for development of software intensive systems was found very effective.
[11]	An assurance methodology to integrate creating of assurance case along with development of system. It was developed to provide identity management, trust and assurance of privacy to the evolving European project PICOS.
[12]	Development Assurance Level (DAL) allocation process based on constraint satisfaction problems (CSP). The CSPs can be solved using developed tool DALculator.
[13]	The development process for the High Assurance ABAC Guard (HAAG), implements Attribute-based Access Control (ABAC) for information requests, and enforces content-based protection and release policies.
[14]	The D*Framework method that can make assurance case for open system and the developed tool D-Case editor support creation of dependability case.
[15]	An assurance directed design approach based on a layered assurance arguments development for developing assurance cases for design of cyber physical system.

III. CONCLUSION AND FUTURE WORK

A Literature review is done on available software design assurance techniques and approaches to understand different methodologies used to have design assurance for the software under development. It is found that Assurance Case based software design approach is more effective and promising in assurance driven design for software systems. It is also found that graphical presentation of an assurance case can be done by using Goal Structuring Notation or by using Claims-Assurance-Evidence (CAE) method. The assurance cases along with the arguments and evidences can be effectively serve the purpose of documentary proof in case required for any verification and validation purpose.
Suggestions for future work include the development of software for any real life practical system using Assurance Case based approach and verify the advantages and areas of improvements if any, offered by this software design methodology.

REFERENCES

[1] Bishop P., Bloomfield R., Guerra s., The future of goal-based assurance cases, Conference proceedings for International Conference on Dependable Systems and Networks, 2004
[2] Scott AT, Krombolz A.H., Structured Assurance Cases: Three Common Standards, High-Assurance Systems Engineering, pp. 99 - 108, 2005
[3] Emmet, L., Guerra, S., Application of a Commercial Assurance Case Tool to Support Software Certification Services, Software Certificate Management Workshop, ASE Conference, 2005
[4] Graydon P.J.,Knight J.C., Strunk E.A., Assurance Based Development of Critical Systems, Dependable Systems and Networks, 2007
[5] Nguyen E.A., Greenwell W.S., Hecht M.J.,Using an Assurance Case to Support Independent Assessment of the Transition to a New GPS Ground Control System, Conference proceedings for Dependable Systems and Networks With FTCS and DCC, IEEE Publication, June 2008
[6] Hall J.G., Rapanotti L., Assurance-driven design in Problem Oriented Engineering, International Journal On Advances in Systems and Measurements, vol. 2, no. 1, pp. 119-130, 2009
[7] Gandhi R.A., Lee S., Assurance Case Driven Case Study Design for Requirements Engineering Research, 15th International Working

Conference, Requirements Engineering: Foundation for Software Quality, pp. 190-196, 2009

[8] Jee E., Lee I., Sokolsky O., Assurance Cases in Model-Driven Development of the Pacemaker Software, 4th International Symposium On Leveraging Application of Formal Methods Verification and Validation (ISoLA), Part II, pp. 343-356, 2010

[9] Graydon P.J., Knight J.C., Process Synthesis in Assurance Based Development of Dependable Systems, Proceedings of the Eighth European Dependable Computing Conference, IEEE Publication, pp. 75-84, 2010

[10] Stringfellow, M.V., Leveson N.G., Owens B.D., Safety-Driven Design for Software-Intensive Aerospace and Automotive Systems, Proceedings of IEEE, vol.98, issue 4, pp.515-525, 2010

[11] José Luis Vivas, Isaac Agudo, Javier Lopez, A Methodology for Security Assurance Driven System Development, Requirements Engineering - Special Issue on Digital privacy: theory, policies and technologies, vol. 16, issue 1, pp. 55-73, 2011

[12] P. Bieber, R. Delmas, and C. Seguin, DALculus: theory and tool for development assurance level allocation, Proceedings of the 30th international conference on Computer safety, reliability, and security, SAFECOMP'11, pp. 43–56, 2011

[13] Wrona K., Oudkerk S., Hein C., Menz N., Ritter T., High-Level Design Process For NATO High Assurance ABAC Guard, Symposium on Architecture Definition and Evaluation, 2013

[14] Saruwatari T., Yamamoto S., Definition and application of an assurance case development method, Springerplus Open Journal, vol. 2, issue 1, 2013

[15] Devesh Bhatt, Gabor Madl, David Oglesby, Sam Owre, Natarajan Shankar, and Ashish Tiwari, Assurance-Directed Design of Cyber-Physical Systems, National Workshop on Transportation Cyber-Physical Systems, 2014