Assurance Case Patterns On-line Catalogue

Monika Szczygielska[1] and Aleksander Jarzębowicz[2(✉)]

[1] Ośrodek Badawczo-Rozwojowy Centrum Techniki Morskiej S.A., Dickmana 62, 81-109 Gdynia, Poland
monika.szczygielska@ctm.gdynia.pl
[2] Department of Software Engineering, Faculty of Electronics, Telecommunications and Informatics, Gdańsk University of Technology, Narutowicza 11/12, 80-233 Gdańsk, Poland
olek@eti.pg.gda.pl

Abstract. Assurance case is an evidence-based argument demonstrating that a given property of a system (e.g. safety, security) is assured. Assurance cases are developed for high integrity systems, as in many industry domains such argument is explicitly required by regulations. Despite the fact that each assurance case is unique, several reusable argument patterns have been identified and published. This paper reports work on development of an on-line assurance case patterns catalogue available in NOR-STA web-based software tool. This work included an extensive literature search, critical evaluation of available patterns and selection of most relevant ones, finally translation of selected patterns to their target representation. The paper also describes a validation case study in which an assurance case for medical devices was reviewed and restructured by introducing patterns. The resulting catalogue was published and its 45 patterns can be directly used in assurance cases built using NOR-STA tool.

Keywords: Assurance case · Safety case · Pattern · Catalogue

1 Introduction

Assurance case is "a structured set of arguments and a body of evidence showing that an information system satisfies specific claims with respect to a given quality attribute" [1]. Assurance cases demonstrating properties like safety or security are developed for high integrity systems and in many industry domains they are explicitly required by regulations [2–4]. Despite the fact that each assurance case is unique, several repeatable problems and situations can be identified. Such problems and generalized solutions for them can be described in the form of patterns. Patterns are used in other engineering domains (e.g. design patterns in software engineering [5]) as general reusable solutions to a commonly occurring problems within a given context.

Patterns have been adopted and successfully used for assurance cases [6]. Several reusable patterns have been published, but their descriptions are distributed among many sources, they are not uniform and their direct use in software supporting tools is usually not possible. The work reported in this paper was the implementation of the idea to create a unified pattern catalogue and to publish it in the Internet. The catalogue is supposed to include all relevant patterns, ready to use in a supporting tool.

W. Zamojski et al. (eds.), *Advances in Dependability Engineering of Complex Systems*, Advances in Intelligent Systems and Computing 582, DOI 10.1007/978-3-319-59415-6_39

In Sect. 2 we outline the background of our research, which includes two main parts: in Sect. 2.1 we describe assurance cases, their usage and main notations; in Sect. 2.2 we provide brief outline of patterns concept and their application to assurance cases. Section 3 presents our contribution: the development and validation of pattern catalogue. In Sect. 4 we discuss conclusions and possible directions of future work. No separate "Related Work" section is defined, to avoid redundancy, as catalogue development (Sect. 3) included a literature review.

2 Background

2.1 Assurance Cases

Assurance cases are developed for high integrity systems to demonstrate that their particular quality attributes are achieved. The most common are safety cases [7] which provide arguments that a system is acceptably safe to operate i.e. will not result in harm to its environment. Other kinds of assurance cases include e.g. security cases, reliability cases or maintainability cases [8].

A growing demand for assurance cases can be observed in several industry domains. Regulations for automotive [2], railway [3], healthcare [4] explicitly require or strongly recommend issuing an assurance case for high integrity systems to be approved by a regulatory body. Also, assurance cases were recently addressed by recognized standards issued by ISO/IEC [9] (later adopted by IEEE) and by OMG [10].

Assurance case development begins with defining high-level claims about system's quality attribute(s). Then a supporting argument is provided. Such argument will include its own, more detailed, lower-level claims, which in turn need to be supported. When necessary, evidence is referenced in the argument.

As a simplified example, consider a top-claim stating that "*A system is safe to operate in its environment*". The supporting argument could include sub-claims that "*Hazard identification activity uncovered all potential hazards*" and that "*All hazards have been eliminated*". "*All hazards have been eliminated*" is further decomposed to sub-claims addressing particular hazards: "*Hazard A is eliminated*", "*Hazard B is eliminated*". Evidence referenced in such argument would include description of hazard identification process, resulting list of hazards, system design documentation etc.

This example is very simple, while the real high integrity systems are usually complex, include a number of components and integrate parts engineered using different technologies. As result, an assurance case for such system is also very complex and supported by a large number of evidence sources. Development and maintenance of real-life assurance cases require suitable ways of expressing argument structures and software tools providing adequate support. To address such needs, dedicated assurance case notations were designed, which allow to express the structure of assurance argument with all essential aspects. On a closer look, the above example lacks many important details e.g.: what is the context of a given claim (e.g. how a "hazard" is defined) or what argumentation strategy is used and what is the rationale behind it (e.g. why are the claims about identification of hazards and their elimination sufficient to argue that the system is safe). Assurance case notations capture such issues as its elements/building blocks.

The notations currently used include CAE (Claim-Argument-Evidence) [11], GSN (Goal Structuring Notation) [12] and NOR-STA [13].

Our work is part of the research done at Gdańsk University of Technology. This research on assurance cases (dated back to 2001 [14]) resulted in TRUST-IT methodology for assurance case development. NOR-STA notation depicted in Fig. 1 is the main component of TRUST-IT (arrows from A to B denote that element A can support element B). Another result is NOR-STA tool, an Internet based software which supports development and maintenance of assurance cases. We use both NOR-STA notation and tool in our work reported in the next sections.

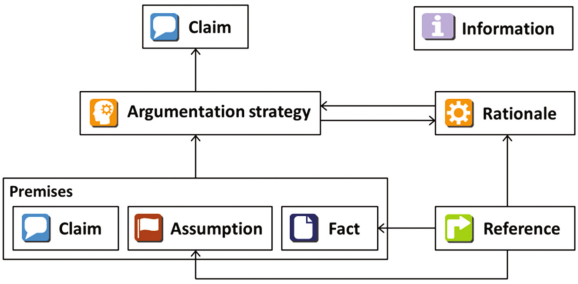

Fig. 1. NOR-STA notation metamodel

2.2 Patterns

Patterns are supposed to enable reuse of successful approaches by describing repeatable design problems and their generic (free of detail) solutions. Patterns were introduced to software engineering domain [5], where they became one of the most important ideas. Software design patterns capture repeatable problems related to object-oriented design and their optimal (with respect to flexibility, ease of maintenance etc.) solutions. According to [5], pattern's description should include the following elements:

Pattern Name, Intent, Also Known As, Motivation, Applicability, Structure, Participants, Collaborations, Consequences, Implementation, Example Applications, Known Uses, Related Patterns

Experiences from software design were adopted to assurance cases, based on the observation that repeatable problems and good practices can be identified for arguments as well. E.g. when demonstrating the safety of the system by using hazard analysis results (as in example from Sect. 2.1), an appropriate hazard decomposition pattern can be used. The first idea of assurance case patterns were described in [15] and the first catalogue of patterns was proposed in [6]. In the following years the concept and applications of patterns were further elaborated and more patterns (as well as catalogues grouping them) were published. Moreover, patterns were accepted by practitioners and currently are widely used in various industry domains [16–18].

3 Assurance Case Patterns Catalogue

3.1 Motivation

The main idea behind our activities was to develop and publish a catalogue summarizing the current state of research and practice on assurance case patterns. According to our prior knowledge from research on assurance cases, further confirmed by an initial literature review, the situation in assurance case patterns area was as follows:

- Several pattern catalogues existed (e.g. [6, 19]), but each of them included a set of distinct patterns (very few patterns shared), and some were dedicated to a more specific problems (e.g. systems built from existing software components [19]).
- Additional patterns, not included in any catalogue but described in separate papers or reports (e.g. [17, 20]) were designed. Such sources described a single pattern or a small number of interrelated patterns.
- Different notations were used for published patterns. The dominating notation used was GSN, but there were exceptions. Furthermore, pattern descriptions used different structures and some provided significantly more details than others.
- The published patterns were usually available only as the contents of a paper, thesis or report. Even if the author designed a pattern using a supporting software tool, such electronic pattern representation was not available to the general research community. Someone who intended to use a given pattern in his/her case, would have to manually enter it, element after element, into a software tool.
- Finally, as far as we know, in recent years no extensive search for existing patterns was performed and no summary about the current state published, except [21] where several sources and patterns known to authors are listed.

These observations motivated us to start the work aimed at development of a single pattern catalogue, summarizing the current state of this research area. We intended to represent those patterns in NOR-STA tool, so they could be published in the Internet and ready to be applied to assurance cases created in NOR-STA. In the next sections we describe the process of catalogue development, the resulting catalogue and the validation case study conducted to assess its applicability to support the work on assurance cases.

3.2 Catalogue Development Process

The first step to develop a pattern catalogue was to conduct a literature search and to identify sources where assurance case patterns are described. We cannot claim it was a Systematic Literature Review, as no review protocol was prepared and no meta-analyses conducted, but we made an effort to make the search as extensive as reasonably possible and to include all possible candidate sources. The main tool used for this purpose was Google Scholar (GS). Our experiences from past literature searches have shown it to be sufficient, as GS indexes other publication databases. This was also confirmed during this search – we reached, through GS, publications stored in other databases e.g. IEEE, Springer or Elsevier. We conducted the search by:

- Using appropriate keywords in GS search engine;
- Using citation maps for sources already found or known beforehand – for each source we checked its References sections and we used GS to identify publications which in turn cite this source;
- Using names of authors of previously identified publications in GS and in a generic web search engine.

The candidate sources found were analyzed by reviewing their titles, keywords and abstracts, also by quickly scanning the contents - pattern definitions are usually represented as figures. After rejecting the sources which clearly included no patterns, we still had 31 sources for a more thorough analysis. Due to space limitations we cannot list all of them in this paper, however such list of references is available in our on-line catalogue. The analysis of sources' contents resulted in several observations:

- Some of the sources, despite using word "pattern", just reported arguments used in development of a particular assurance case, not patterns understood as more generic, well-designed solutions, applicable to many assurance arguments.
- Other sources proposed patterns dedicated to a single domain (e.g. automotive [17] or nuclear [18]) or at least it was not clear whether they could be adopted to another domain (and if so, how should such pattern be modified/generalized).
- In a very few cases, a pattern appeared in more than one catalogue (e.g. ALARP Pattern in [6, 22]).
- When comparing patterns, some of them could be treated as more generalized versions of others (e.g. Architectural Decomposition pattern from [23] is more general than the corresponding patterns from [17]).

We made a selection among "candidate patterns" by applying the following actions:

- Reject the particular arguments not generalized into patterns.
- Reject patterns described as ideas only, without explicit argument structure.
- If a given pattern is included in multiple catalogues (or other sources) – select the most recent source.
- If similar argumentation structures are described as patterns – select the more general one.
- Reject domain-specific patterns, for which no indication is provided how to apply them to other domains.

As result, we selected 45 patterns. The most difficult decision concerned COTS Safety Patterns [19], we finally decided to include two core patterns and leave out the remaining ones.

The next steps were to translate these patterns into NOR-STA notation, to provide a uniform description structure for them and to represent them in NOR-STA tool. As all the selected patterns were defined in GSN notation, we had to define translation rules between GSN and NOR-STA notations. The translation encountered no fundamental problems, however some difficulties stemming from notational differences were uncovered e.g.:

- In GSN a Goal (an equivalent of a Claim) can be directly supported by a sub-goal. In NOR-STA an Argumentation strategy is mandatory between Claims to explain the argument. In such cases we had to add Argumentation strategies.
- In GSN, a Justification is an optional element. In NOR-STA, every Argumentation strategy is expected to have a Rationale, which had to be added.
- No explicit Context element is defined in NOR-STA, instead Information elements are used to express all contextual or explanatory information included in assurance arguments.
- GSN defines additional elements, which are useful for pattern instantiation e.g. marking some parts of an argument as optional or alternative. With no direct equivalent in NOR-STA, we had to use Information elements for this purpose.

As for description structures, there were significant differences between the sources of selected patterns. Sources [6, 19, 24] used a full description structure adopted from software design patterns (listed in Sect. 2.2), [25] covered about a half of description elements, [22, 23] provided only a structure of the pattern, while [20, 26] a structure plus examples of use. All selected patterns were (manually) entered into NOR-STA tool, using translation rules mentioned above. All description elements defined for a given pattern were preserved. Moreover, for each pattern we provided a link to the source of its original description.

Figure 2 is a screenshot from NOR-STA tool depicting an example pattern (Security Case Pattern) as represented in our catalogue. Table 1 gives the summary of all patterns included in our catalogue and their sources. The catalogue is available on-line at [27] (please note that read-only access is enabled for unauthorized users). The total number of basic elements the included patterns are composed of is over 1300.

3.3 Validation Case Study

To assess whether the pattern catalogue is applicable in typical activities related to assurance case development and maintenance, we conducted a validation case study. Introducing patterns as part of maintenance activities (updates and modifications applied to already developed assurance case) is more demanding than using patterns during the development. In development, the author introduces an "empty" pattern while building a part of assurance case and instantiates it (fills with system-specific content). In maintenance, additionally a part of assurance case has to be restructured, some arguments modified, new elements added, while others relocated – and all that without losing any existing essential information and in such way that the resulting overall assurance case is valid and convincing.

We decided to use an existing assurance case dedicated to safety and operability of medical devices, developed by Kansas State University as part of the project commissioned by U.S. Food and Drug Administration (FDA) [28]. It was not a case for a specific product. Its purpose was to be a generic assurance case for a Patient Control Analgesia (PCA) pump serving as an example for infusion pump manufacturers, who, according to recent guidelines [4], are expected to deliver to FDA assurance cases for their products. This case was developed using NOR-STA and is freely available at [29].

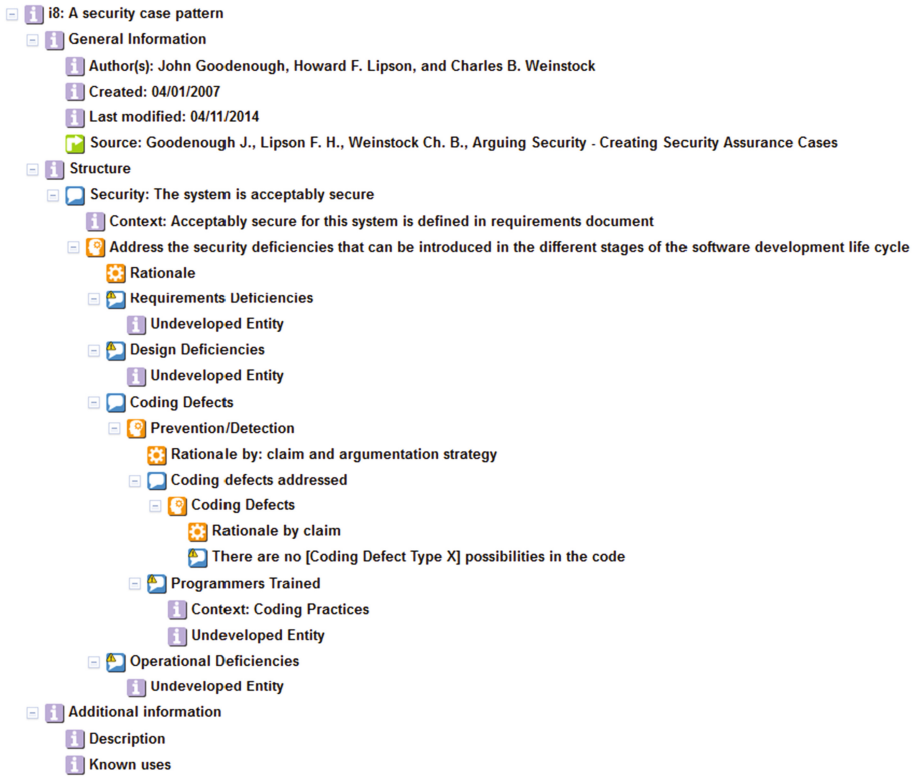

Fig. 2. Security Case Pattern represented in NOR-STA tool.

Being a generic example, the case was not complete e.g. several lower-level claims depending on device's design decisions remained undeveloped. The case reflected FDA's requirements e.g. to address all hazard categories (electrical, software, bio-chemical etc.), to show that remaining risk after mitigation of all hazard can be considered acceptable etc. The resulting argument was quite large as it included about 750 elements. No patterns were explicitly used in it. In the case study:

- On the methodological level, we intended to verify whether the patterns from our catalogue are applicable for a maintenance of a non-trivial assurance case;
- On the operational level, we wanted to find out how easy/difficult can such operation be done using NOR-STA tool functionality.

One of us analyzed the PCA pump assurance case and identified the parts where patterns could be introduced. The results were reviewed by a second person and resulted in some change proposals. After agreement was reached, a new version of the assurance case was created by copying patterns from the catalogue, pasting them to a

given part of the case and restructuring this part, so the pattern became an integral part of argument structure and was filled with specific contents. Table 2 provides a summary about patterns used and the parts of assurance case they were introduced to.

Table 1. Patterns included in the catalogue and their sources

Sources	Patterns
Safety Case Patterns Catalogue (T. Kelly) [6]	ALARP (As Low as Reasonably Practicable) Argument
	Hazard Directed Integrity Level Argument
	Control System Architecture Breakdown Argument
	Diverse Argument
	Safety Margin
	Fault Tree Evidence
A Software Safety Pattern Catalogue (R. Weaver) [24]	Component Contributions to System Hazards
	Hazardous Software Failure Mode Decomposition
	Hazardous Software Failure Mode Classification
	Software Argument Approach
	Absence of Omission Hazardous Software Failure Mode
	Absence of Commission Hazardous Software Failure Mode
	Absence of Early Hazardous Software Failure Mode
	Absence of Late Hazardous Software Failure Mode
	Absence of Value Hazardous Software Failure Mode
	Handling of Hardware/Other Component Failure Mode
	Effects of Other Components
	Handling of Software Failure Mode
The Software Safety Argument Pattern Catalogue (R. Hawkins, T. Kelly) [25]	High-Level Software Safety Argument Pattern
	Software Contribution Safety Argument Pattern
	SSR Identification Software Safety Argument Pattern
	Hazardous Contribution Software Safety Argument Pattern
	Software Contribution Safety Arg. Pattern with Grouping
Safety Cases for Advanced Control Software: Safety Case Patterns (J. McDermid, R. Alexander, T. Kelly, Z. Kurd) [22]	Improved Safety Argument
	Maintained Safety Argument
	At Least As Safe Argument
	Risk Acceptance Argument
	Top Level System-to-Software Hazard Mitigation Argument
	Top Level System-to-Software Hazard Contribution Arg.
	Software Hazard Contributions Argument
	Hazardous Software Failure Mode Acceptability Argument
	Hazardous Software Failure Mode Absence Argument
	Safe Adaptation Argument
	Behavioural vs. Model-Building Adaptation Argument

(*continued*)

Table 1. (*continued*)

Sources	Patterns
COTS Safety Patterns (F. Ye) [19]	COTS Component Use Safety Argument
	Process-Based COTS Safety Argument
Decomposition Patterns (S. Yamamoto) [23] – conference presentation slides	Architecture Decomposition
	Functional Decomposition
	Attribute Decomposition
	Infinite Set Decomposition
	Complete Decomposition
	Monotonic Decomposition
	Decomposition by Concretion
Arguing Security (Weinstock et al.) [26]	A Security Case Pattern
Model-Based Develop-ment (Ayoub et al.) [20]	From_to Pattern

Table 2. Case study summary

Assurance case part	Pattern introduced
Arguing system safety by addressing pre-defined categories of hazards	Component Contributions to System Hazards [24]
Mitigation of "Incorrect flow rate" hazard by providing built-in alarms	Monotonic Decomposition [23]
Arguing PCA pump performs intended function on the basis or valid specification and correct implementation	from_to [20]

Several more patterns could be used, but they would affect the same parts as those listed in Table 2, therefore it was a choice between alternatives (e.g. Component Contributions to System Hazards [24] and Architectural Decomposition [23]). Introduction of patterns improved the assurance case, for example "from_to" pattern required adding definitions of intended use and intended environment, which were not explicitly expressed in the original case.

4 Conclusions and Further Work

The work reported in previous sections resulted in a catalogue of assurance case patterns grouping 45 patterns gathered from available literature. Of course, more patterns could be included, however it was our explicit decision to be selective and take into consideration only the universal, not domain-specific ones. The catalogue is the end result of an extensive literature overview. Together with our working materials and reference lists it can be considered a snapshot of assurance case patterns research & practice state in a given moment of time. Our catalogue is available to any Internet user.

This also serves as a way to disseminate knowledge about existing patterns. The registered NOR-STA users can utilize it by copying and manually instantiating patterns of their choice in their own assurance cases. The feasibility of such operations was validated in the performed case study.

In future, we plan to introduce automated pattern instantiation in NOR-STA tool. Basically, it means implementing software tool functionality which allows user to select a pattern and a source of data necessary to fill the contents of such pattern and then the instantiation is done by the tool. For example, it could be a pattern related to hazards (like ALARP Pattern [6]) and an external file storing hazard analysis results in a specified format. It is currently a subject of active research at Gdańsk University of Technology, which already resulted in elaborating automated instantiation method and first working software prototype [30]. The catalogue can be easily extended with additional patterns and we intend to do it as new patterns appear in the literature. Also, domain-specific patterns, which were rejected during catalogue development process, can be added in future (if for example there is such demand from NOR-STA users) or included in separate, domain-specific catalogues.

References

1. Kissel, R.: Glossary of key information security terms. Revision 2, NIST IR 7298. National Institute of Standards and Technology (2013)
2. International Organization for Standardization (ISO): ISO/DIS 26262: Road Vehicles - Functional Safety (2011)
3. CENELEC: EN 50126. Railway Applications: The Specification and Demonstration of Reliability, Availability, Maintainability and Safety (RAMS) (1999)
4. FDA: Infusion Pumps Total Product Life Cycle, Guidance for Industry and FDA staff (2014)
5. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, Reading (1995)
6. Kelly, T.: Arguing safety – a systematic approach to safety case management. Ph.D. thesis, Department of Computer Science, University of York (1998)
7. Maguire, R.: Safety Cases and Safety Reports: Meaning, Motivation and Management. Ashgate Publishing Ltd, Aldershot (2006)
8. Despotou, G., Kelly, T.: Extending the safety case concept to address dependability. In: Proceedings of 22nd International System Safety Conference, pp. 645–654 (2004)
9. International Organization for Standardization (ISO): 15026-2:2011: Systems and Software Engineering – Systems and Software Assurance – Part 2: Assurance Case (2011)
10. Object Management Group: Structured Assurance Case Metamodel ver. 1.1 (2015)
11. Adelard: Claims, Arguments and Evidence (CAE). http://www.adelard.com/asce/choosing-asce/cae.html
12. GSN Community Standard Working Group: GSN community standard version 1 (2011). http://www.goalstructuringnotation.info/
13. Argevide: NOR-STA Argument Notation White Paper. https://www.argevide.com/sites/default/files/docs/Argevide%20WP2%20-%20NOR-STA%20argument%20notation.pdf
14. Górski, J., Jarzębowicz, A., Leszczyna, R., Miler, J., Olszewski, M.: An approach to trust case development. In: Proceedings of the 22nd International Conference on Computer Safety, Reliability and Security (SAFECOMP 2003). LNCS, vol. 2788, pp. 193–206 (2003)

15. Kelly, T., McDermid, J.: Safety case construction and reuse using patterns. In: Proceedings of SAFECOMP 1997, pp. 55–69 (1997)
16. Hawkins, R., Clegg, K., Alexander, R., Kelly, T.: Using a software safety argument pattern catalogue - two case studies. In: Proceedings of the 30th International Conference on Computer Safety, Reliability and Security (SAFECOMP 2011). LNCS, vol. 6894, pp. 185–198 (2011)
17. Khalil, M., Schätz, B., Voss, S.: A Pattern-based approach towards modular safety analysis and argumentation. In: Proceedings of ERTS 2014, Toulouse, France. LNCS, vol. 8822, pp. 137–151 (2014)
18. Hauge, A., Stølen, K.: A pattern-based method for safe control systems exemplified within nuclear power production. In: Proceedings of the 31st International Conference on Computer Safety, Reliability and Security (SAFECOMP 2012). LNCS, vol. 7612, pp. 13–24 (2012)
19. Ye, F.: Justifying the use of COTS components within safety critical applications. Ph.D. thesis, Department of Computer Science, University of York (2005)
20. Ayoub, A., Kim, B., Lee, I., Sokolsky, O.: A safety case pattern for model-based development approach. In: Proceedings of the 4th NASA Formal Methods Symposium (NFM 2012). LNCS, vol. 7226, pp. 141–146 (2012)
21. Denney, E., Pai, G.: Safety case patterns: theory and applications. NASA/TM–2015–218492 Technical report (2015)
22. Alexander, R., Kelly, T., Kurd, Z., McDermid, J.: Safety cases for advanced control software: safety case patterns. Technical report, University of York (2007)
23. Yamamoto, S., Matsuno, Y.: An evaluation of argument patterns to reduce pitfalls of applying assurance case. In: Proceedings of 1st International Workshop on Assurance Cases for Software-Intensive Systems (ASSURE 2013), pp. 12–17 (2013)
24. Weaver, R.: The safety of software – constructing and assuring arguments. Ph.D. thesis, Department of Computer Science, University of York (2003)
25. Hawkins, R., Kelly, T.: A software safety argument pattern catalogue. Technical report, University of York (2013)
26. Weinstock, C., Lipson, H., Goodenough, J.: Arguing security - creating security assurance cases. US CERT BSI (Build Security In) report, Carnegie Mellon University (2007)
27. Assurance Case Patterns On-line Catalogue, Gdańsk University of Technology. http://www.nor-sta.eu/en/en/news/assurance_case_pattern_catalogue
28. Larson, B.R., Hatcliff, J., Chalin, P.: Open source patient-controlled analgesic pump requirements documentation. In: 5th International Workshop on Software Engineering in Health Care (SEHC), pp. 28–34 (2013)
29. Larson, B.R.: Open PCA Pump Assurance Case, Santos Research Group, Kansas State University (2014). http://openpcapump.santoslab.org/
30. Wardziński, A., Jarzębowicz, A.: Towards safety case integration with hazard analysis for medical devices. In: Proceedings of 4th International Workshop on Assurance Cases for Software-intensive Systems (ASSURE 2016). LNCS, vol. 9923, pp. 87–98 (2016)