This is a preprint of a paper presented at
4th International Workshop on Assurance Cases for Software-intensive Systems (ASSURE 2016), 20 Sept. 2016, Trondheim, Norway
and published in
"Computer Safety, Reliability, and Security", Vol. 9923 of the series Lecture Notes in Computer Science, pp. 87-98.
Final version available at: https://doi.org/10.1007/978-3-319-45480-1_8

Towards Safety Case Integration with Hazard Analysis for Medical Devices

Andrzej Wardziński[1,2] and Aleksander Jarzębowicz[1,2]

[1] Department of Software Engineering, Faculty of Electronics, Telecommunications and Informatics, Gdańsk University of Technology, Narutowicza 11/12, 80-233 Gdańsk, Poland
{andrzej.wardzinski, olek}@eti.pg.gda.pl
[2] Argevide sp. z o.o., Poland
{andrzej.wardzinski, aleksander.jarzebowicz}@argevide.com

Abstract. Safety case is one of system safety lifecycle products and should be consistent with other lifecycle products like hazard analysis results. In this paper we present a method of safety case integration with hazard tables based on the use of parametrized argument patterns. We describe a hazard table metamodel, a safety argument pattern and a mechanism of pattern instantiation using a linking table which represents references to system lifecycle artefacts. We report and comment results of a feasibility study of pattern application for medical device hazard analysis. Finally we discuss the opportunities of applying such solution to safety case development and maintenance and the perspectives of further development of this approach.

Keywords: safety case, hazard table, safety argument pattern, infusion pump, medical device

1 Introduction

A safety case is a way of arguing system's safety used in many industry sectors. In recent years a growing interest in application of safety cases in healthcare can be noticed [1-3]. Such interest is also reflected in regulatory requirements, in particular U.S. Food and Drug Administration (FDA) published a guidance document for manufacturers of medical devices (infusion pumps), strongly recommending delivering safety cases as a part of pre-market notification [4]. It is expected that the safety case approach will be extended for other medical devices in the coming years. The mentioned guidance is complemented by other documents which address other safety-related aspects of medical devices like software components [5] and security [6].

Safety cases are usually based on the results of hazard analysis. FDA recommends tabular form of hazard analysis results presentation for medical devices containing software premarket notifications [5]. The recommended standard describing the process of hazard analysis for medical devices is ISO 14971 [7], which does not impose any particular form of hazard analysis results presentation. Tabular presentation is described by Jones and Taylor [8], who present an idea of transforming hazard tables

into instantiations of argument patterns to be included in a safety case. They also provide an example of a generic pattern.

Our goal is to develop a method to establish and maintain relationship between safety case elements and hazard analysis results through the pattern instantiation process. We use NOR-STA tool [9-10] to develop safety cases and argument patterns. The tool allows to save the safety argument in XML format conformant to OMG SACM standard [11]. The approach we present is based on processing XML data for safety cases and hazard analysis.

In section 2 we present the background and related work including safety cases for medical devices, safety argument patterns and pattern instantiations. In section 3 we describe the metamodel of hazard table and the safety argument pattern mapping to hazard table elements. A case study of the instantiation process is presented in in section 4. The achieved results and future work is discussed in Section 5. In Section 6 we discuss the main conclusions to summarize the presented work.

2 Background

The work presented in this paper concerns safety cases for medical devices, safety case patterns and pattern instantiation.

2.1 Safety cases for medical devices

Safety cases (or assurance cases as referred to in many papers) are a relatively new tool for managing safety of medical devices. One of the first research reports on safety cases for medical devices was published in 2009 by Weinstock and Goodenough [12]. They presented the example of an assurance case for the generic infusion pump and discussed the applicability of assurance case approach for medical devices, especially in the context of FDA's review processes. Ray and Cleaveland [13] introduce an approach to the creation of assurance cases for pre-market submissions of medical devices. It includes argumentation schemes of addressing hazards and providing mitigation mechanisms. Wassyng et al. [14] propose capturing the requirements of a standard (or a guideline) in the form of an assurance case template. As already mentioned, Jones and Taylor [8] designed a safety argument pattern using data from hazard tables documenting risk analysis process for a medical device.

A large repository of safety-related resources for medical devices can be found at the Generic Infusion Pump Research Project website [15]. A number of contributions from University of Pennsylvania was dedicated to several aspects of assurance cases for medical devices e.g. a pacemaker assurance case [16], from_to pattern [17] or a high-level safety argument for the PCA closed-loop system [18]. Also, Larson developed a draft assurance case for Open PCA infusion pump as an example to illustrate how to apply FDA guidelines [19].

2.2 Safety Argument Patterns

The first ideas of safety argument patterns and their role in development of safety cases were described by Kelly and McDermid in [20] and [21]. The first catalogue of patterns was included in Kelly's PhD thesis [22]. In the following years the concept and applications of patterns were further elaborated, mostly by the researchers affiliated with the University of York (e.g. [23, 24]).

A number of pattern catalogues was published over the years: [22][25-28]. Recently, Denney and Pai summarized the existing catalogues and provided a description of six new patterns [29]. An online pattern catalogue including a substantial set of patterns derived from the available sources has been published by Gdańsk University of Technology in NOR-STA tool [30].

The process of pattern application is called instantiation and it requires to define values for pattern parameters, which are specific for a given system [22]. Hauge and Stølen [31] introduce a pattern-based method, called Safe Control Systems (SaCS), which focuses on pattern compositions (integrating sets of patterns) and their instantiations. Khalil et al. [32] describe a reusable pattern library for automotive safety cases and the mechanism for their instantiation.

Denney and Pai [29] provide a formalized definition of safety argument patterns which includes aspects of their instantiation. The mechanism of patterns instantiation consists of an algorithm and data tables, which store traces between template elements and their instantiations. The mechanism was implemented in AdvoCATE tool [33]. The presented instantiation requires interaction from the user of the tool, who is supposed to provide concrete values for pattern parameters. The earlier paper of the same authors [34] focuses on assembling parts of a safety case on the basis of external artefacts in tabular form: hazard tables and two kinds of requirements tables. Two argument patterns for representing contents of hazard tables and requirements tables are proposed. The contents of a hazard table and the structure of corresponding argument pattern are specific to NASA standards and guidelines.

Hawkins et al. [35] present a way of pattern instantiation using a weaving model, which is the main source of information for the instantiation program. The weaving model stores the dependencies between the elements of safety argument patterns and reference information metamodels, as well as additional interdependencies. Reference information models of various notations and tools, based on different metamodels (e.g. system components, errors) can be used to provide values for pattern parameters.

3 Safety Case to Hazard Table Relationship

Safety cases refer to hazards, their causes and control measures. Our work is based on the hazard table format specified in [8] which includes the following table columns:

- Hazardous situation – circumstances in which people, property or environment are exposed to a hazard;
- Causes of the hazardous situation – events and circumstances necessary to the occurrence of hazardous situation;

- Risk estimation before mitigation or severity of harm – risk arising from a hazardous situation, calculated on the basis of probability of occurrence and severity of consequences (or just severity if probability cannot be assessed);
- Control measure(s) – mechanisms applied by the manufacturer to reduce unacceptable risk by addressing causes of the hazardous situation;
- Safety decision rationale – justification why a control measure is chosen and considered to be effective;
- Verification of effectiveness (methods & objective evidence) – verification whether control measure is effective in the context of design specifications and expected behavior;
- Verification of implementation & objective evidence (validation) – validation whether the control measure is fit for purpose in the context of device intended use.

We have specified a hazard table metamodel in the form of an UML class diagram to precisely specify hazard analysis artefacts and their relationship (Fig. 1).

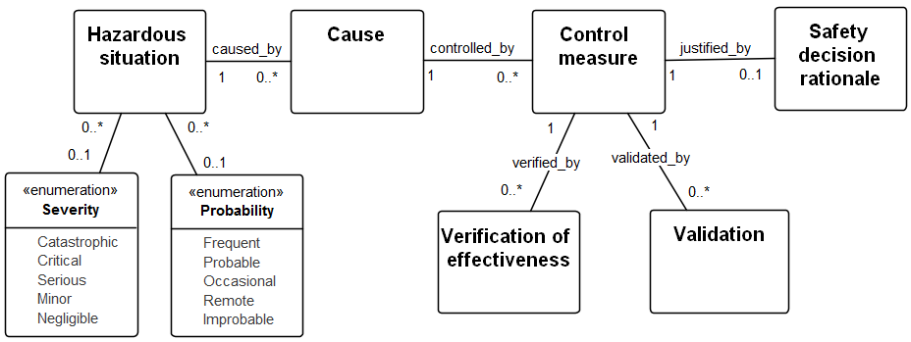

Fig. 1. A metamodel of hazard table elements and their relationships.

The model in Fig. 1 corresponds to top-down approach. We accept optional relationships (0..*) to address situations when hazard analysis is still in progress and is not complete (for example control measures are not yet defined for a given cause). On the other hand we do not accept low level artefacts (e.g. validation evidence) not connected to any control measure. The hierarchy presented in Fig. 1. can be directly mapped to the safety case argument hierarchy. The mapping is described in Table 1.

Safety argument pattern presented in Fig. 2. is based on this hierarchical relationship. The pattern is expressed in textual hierarchical notation used in NOR-STA software tool [36]. The notation is compatible to OMG SACM and includes its main concepts. The types of the elements are denoted by icons and by mnemonics: C – claim, A – argumentation strategy, F – fact, R – rationale, Ctx – context. Argument elements related to hazard table columns are marked with a corresponding column number (ID) specified in Table 1. One should note that NOR-STA notation does not currently implement structural abstraction relations such as multiplicity and choice. Temporary solution presented in Fig. 2 is to describe the relation in UML-like style: “1..*”. NOR-STA notation is planned to be extended to cover structural abstraction.

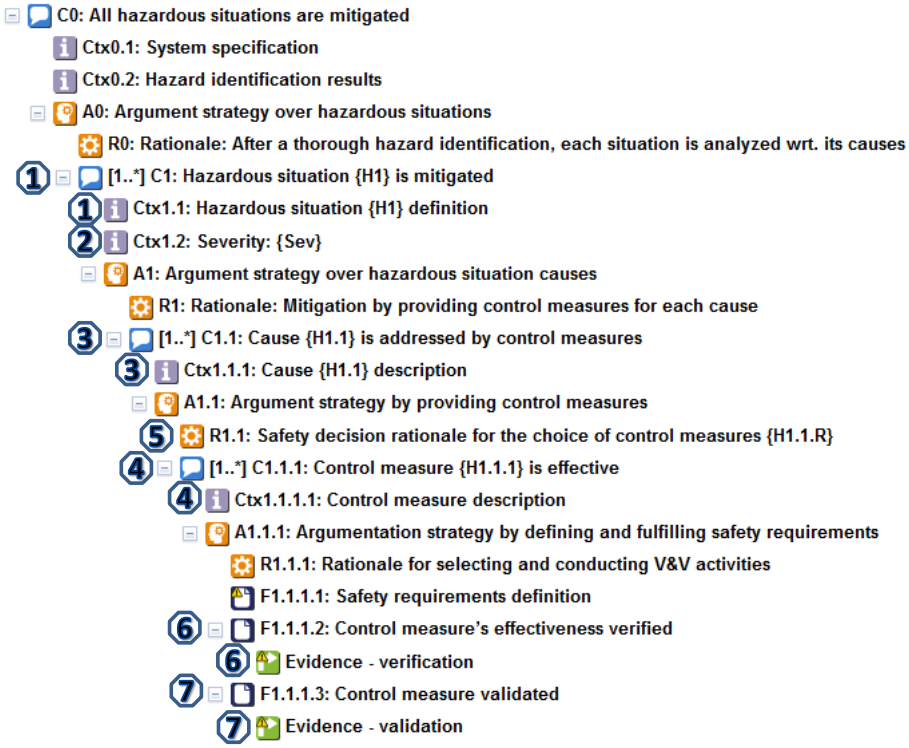

Fig. 2. A structure of a safety argument pattern based on hazard table.

Table 1. Mapping between a hazard table and a safety case.

ID	Hazard table column	Safety case element
1	Hazardous situation	Claim: Hazardous situation is mitigated
		Context: Hazardous situation definition
2	Risk estimation/severity of harm	Context: Severity
3	Causes of the hazardous situation	Claim: Cause is addressed by control measures
		Context: Cause description
4	Control measure(s)	Claim: Control measure is effective
		Context: Control measure description
5	Safety decision rationale	Rationale: Rationale for the choice of control measures
6	Verification of effectiveness – Methods & objective evidence	Fact: Control measure's effectiveness verified
		Evidence: Verification evidence
7	Verification of implementation & objective evidence (validation)	Fact: Control measure validated
		Evidence: Validation evidence

The presented safety argument pattern is simplified and does not cover issues like: hierarchical hazard decomposition, re-evaluation of the residual risk following application of control measures, mitigation strategies other than addressing causes of hazardous situations. The real safety case would also have to be extended by arguments and evidence demonstrating the confidence in safety claims. For example, one could doubt whether hazard identification uncovered all hazardous situations. Such doubts should be addressed by a separate confidence case or by local confidence arguments supporting Rationale elements [37], in this case a confidence argument for R0 in Fig. 2.

4 Hazard Table Integration with Safety Case

Hazard analysis results mapping to safety case elements can be can be established in the safety case pattern instantiation process. In this section we will present the use of parametrized patterns to integrate safety case and hazard table and to track the relationships. First we will describe safety argument instantiation mechanism and then present how it can be applied to safety case integration with hazard tables.

4.1 Pattern Instantiation Process

The objective of the instantiation process is to produce a safety argument compiled from an argument pattern and references to the artefacts of types specified by pattern parameters. Pattern parameters may refer to any system model or artefact.

Our basic assumption for pattern instantiation process is the use of XML representation for all system models, the safety case and patterns. We introduce a linking table to track relationships between models. The linking table is divided into two parts:

- Abstract part is created for each pattern to specify the type of referenced models and the type of target elements for each pattern parameter. For example we can specify a pattern parameter to be related to a *ControlMeasure* type specified in the hazard object model (Fig. 1).
- Instantiation part defines relationships on detailed system model level. For each pattern parameter a specific model element can be selected by the user or the parameter value is entered manually.

Both parts of the linking table are presented in their context in Fig. 3. During the instantiation process, a user has to select elements of a specified type in the system model. Let's take an example of {H1} parameter in the pattern presented in Fig. 2. The abstract linking table can specify that {H1} parameter is related to objects of *HazardousSituation* type in the hazard table class model (Fig. 1). During the instantiation process, the user will be asked to point to an XML file for hazard table data and then select objects of the *HazardousSituation* type. As a multiplicity operator [1..*] is defined for {H1} parameter in the template, the user will be asked to select any number of objects and an argumentation subtree will be created for each of them.

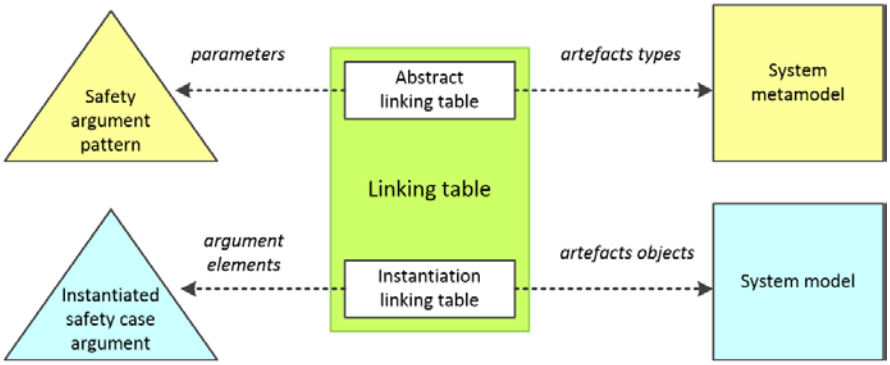

Fig. 3. Linking table and referenced models (arrows show references).

4.2 Integration with Hazard Table Case Study

We will demonstrate the instantiation process for a simple example of a hazard decomposition argument. All the input information used in the process is represented in XML formal, as well as the final output. From a technical point of view it is an XML transformation process. We will present model excerpts in XML format or GSN-like diagrams. NOR-STA tool generates graphical argument diagrams, however some symbols used differ a bit from standard GSN, for example the context elements. We assume the differences will not impede understanding of the diagrams.

The pattern presented in Fig. 4 is a fragment of the pattern from Fig. 2. As XML representation takes much more space than the diagram, we present only a small excerpt containing claim C1 and context Ctx1.2, represented as XML in Fig. 5.

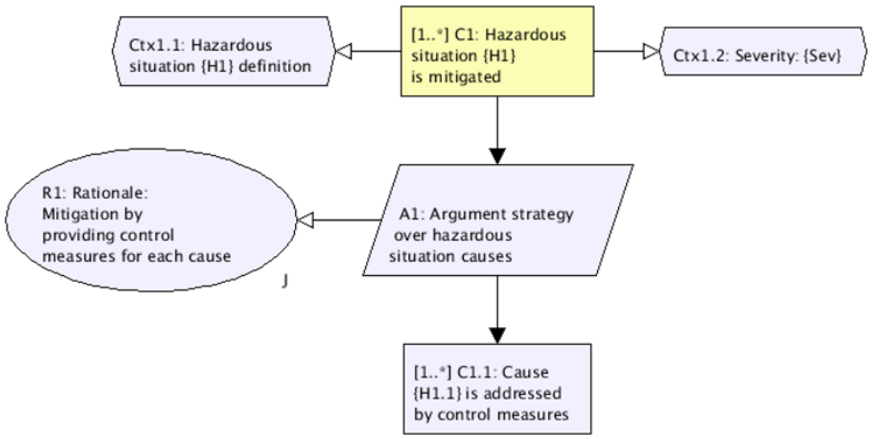

Fig. 4. Safety case pattern excerpt for hazard decomposition by causes.

```
<containsArgumentElement
    content="Hazardous situation {H1} is mitigated"
    identifier="[1..*] C1" xmi:id="7" xsi:type="ARM:Claim">
    <taggedValue key="order" value="1"/>
</containsArgumentElement>
<containsArgumentElement
    content="Severity: {Sev}"
    identifier="Ctx1.2" xmi:id="9" xsi:type="ARM:InformationElement">
    <taggedValue key="order" value="2"/>
</containsArgumentElement>
<containsAssertedRelationship
    source="7" target="9" xmi:id="30" xsi:type="ARM:AssertedContext"/>
```

Fig. 5. XML representation of C1 and Ctx1.1 elements of the pattern from Fig. 4.

There are three parameters in this pattern fragment: hazardous situation {H1}, severity {Sev} and cause {H1.1}. The abstract linking table (Table 2) allows us to map these parameters to hazard analysis metamodel elements (Fig. 1).

Table 2. Abstract linking table.

Pattern parameters		System metamodel	
Pattern name	**Parameter name**	**Model type**	**Element type**
HazardDecomposition	H1	HazardAnalysis	HazardousSituation
HazardDecomposition	Sev	HazardAnalysis	Severity
HazardDecomposition	H1.1	HazardAnalysis	Cause

To instantiate the pattern we will need a hazard model for the system under analysis. Our safety case example refers to PCA infusion pump system [38]. Excerpt of the hazard analysis in XML format is presented in Fig. 6. This fragment describes one cause (sensor failure) for a hazardous situation ‘air in line’. The possible consequence is the injection of air into the patient bloodstream which can be dangerous for patient life and health.

```
<hazardElement
    content="Air in line"
    xsi:type="HA:HazardousSituation" xmi:id="H1">
    <attribute xsi:type ="HA:Severity" xmi:id="S1" name="Critical"/>
    <attribute xsi:type ="HA:Probability" xmi:id="P1" name="Occasional"/>
</hazardElement>
<hazardElement
    content="Sensor does not signal error when air is present in IV line"
    xsi:type="HA:Cause" xmi:id="C1">
</hazardElement>
<relationship
    xsi:type="HA:CausedBy"
    xmi:id="H1C1" source="H1" target="C1">
</relationship>
```

Fig. 6. Model excerpt for one hazardous situation and one of its causes.

During safety argument’s instantiation the user has to select value for each pattern parameter. The value can be an element of the hazard model of appropriate type or the value may be entered manually by the user. The result of this step is recorded in the

instantiation linking table (Table 3). For each parameter, the table specifies corresponding safety case elements (presented in Fig. 7) and system model elements (XML excerpt of a hazard table in Fig. 6),

Table 3. Instantiation linking table.

Parameter name *(abstract linking table)*	**Safety case**		**System model**	
	Pattern root element id	**Elements ids**	**Filename**	**Element id**
H1	C1	C1, Ctx1.1	PCA_hazards.xmi	H1
Sev	C1	Ctx1.2	PCA_hazards.xmi	S1
H1.1	C1	C1.1	PCA_hazards.xmi	C1

The instantiation linking table directly points argument elements to the values of the specified model elements.

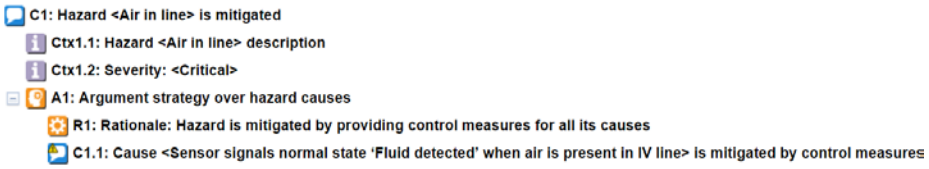

Fig. 7. Excerpt from the instantiated argument pattern.

The instantiation linking table is not deleted after the instantiation as it allows for tracking the relationship even in case the model changes. In case any hazard table element is modified, the change can be propagated to dedicated safety case elements provided objects identifiers are maintained.

5 Summary of the Case Study and Further Work

The presented case study demonstrates how the linking table can be used to establish the lasting relationship between a safety case and a hazard table. The established relationship should be maintained throughout the system lifecycle. The linking table can be used to track and propagate changes. Let's consider a situation when a hazard cause has been modified in the hazard table. Having the linking table filled in, we can detect the change and react to it. When the change is to be propagated to the safety case, we can re-instantiate safety case elements affected by the change or even restart the whole instantiation process and produce new and up-to-date safety case. If we want this process to be effective, we should forbid manual safety argument modifications or limit them to safety case areas not covered by the automatic instantiation process.

Change propagation in the opposite direction is also possible, however we should be careful in allowing changes to be propagated from a safety case to a hazard table. From a technical point of view it will not be difficult to implement a two way change

propagation mechanism. The issue is whether it is necessary and secure to allow changing the safety case without the actual update of the hazard analysis.

Propagation of structural changes is more difficult and will require extending the linking table with additional information. As a structural change we understand adding or removing any model element. For example when a new hazard cause has been identified. Change propagation would require creation of a new argument subtree. And vice versa, when an element is deleted (let's imagine we have to delete one of the hazard causes) from the hazard table, the change propagation mechanism would cause removal of the related argument parts as specified in the linking table.

We plan to extend the linking table to comprehend data necessary for propagation of structural changes in the hazard table. This would enable continuous consistency maintenance between a safety case and a hazard table.

The pattern described in Section 3 bears similarities to Extended Hazard Directed Breakdown Pattern [29], however the latter includes hierarchical decomposition of hazards into lower-level ones. This is possible with the use of loop construct which is not available in NOR-STA notation (as NOR-STA data structure is based on directed graph, not hypergraph). We can use dedicated Link elements to represent loops in NOR-STA notation to achieve the same effect.

6 Conclusions

We presented the approach of integrating safety cases and hazard tables based on the use of parametrized safety argument patterns. The essential concept is the use of the linking table which stores references to the elements of safety case and hazard table both on abstract (pattern parameters, hazard table columns) and instantiation (claims, hazardous situations etc.) levels. On the abstract (pattern) level we map pattern parameters to metamodel elements and then on the instantiation level we map each parameter value to a particular model element. The linking table allows to track the relationships and maintain consistency between the safety case and hazard table.

This approach can be generalized from the hazard table presented in this paper to other system models, provided we can specify a metamodel and provide an XML interface, for example for AADL specifications. The approach can also be applied to other safety argument patterns however the user would need to specify appropriate system models for all pattern parameters.

This paper presents work in progress and the linking table may evolve as the approach matures. The presented approach will be developed further to effective management and maintenance of the relationship between safety cases and hazard tables.

References

1. Sujan M., Koornneef F., Chozos N., Pozzi S., Kelly T.: Safety Cases for Medical Devices and Health IT - Involving Healthcare Organisations in the Assurance of Safety, Health Informatics Journal 19(3), pp. 165-182 (2013)

2. Chen Y., Lawford M., Wang H., Wassyng A.: Insulin pump software certification, in: Foundations of Health Information Engineering and Systems, pp. 87-106, Springer Berlin Heidelberg (2013)
3. Sujan M., Habli I., Kelly T., Pozzi S., Johnson C.: Should healthcare providers do safety cases? Lessons from a cross-industry review of safety case practices, Safety Science 84, pp. 181-189 (2016)
4. FDA: Infusion Pumps Total Product Life Cycle, Guidance for Industry and FDA staff (2014)
5. FDA: Guidance for the Content of Premarket Submissions for Software Contained in Medical Devices, (2005)
6. FDA: Content of Premarket Submissions for Management of Cybersecurity in Medical Devices. Guidance for Industry and Food and Drug Administration Staff (2014)
7. ISO: ISO 14971:2007. Medical devices -- Application of risk management to medical devices (2007)
8. Jones P.L., Taylor A.: Medical Device Risk Management And Safety Cases, Bio-medical Instrumentation & Technology 49 (s1), pp. 45-53 (2015)
9. Górski J., Jarzębowicz A., Miler J., Witkowicz M., Czyżnikiewicz J., Jar P.: Supporting Assurance by Evidence-Based Argument Services, Proc. of SAFECOMP 2012, LNCS 7613, pp. 417-426 (2012)
10. NOR-STA tool website, https://www.argevide.com/en/products/assurance_case
11. OMG: Structured Assurance Case Metamodel (SACM), Version 1.1 (2015)
12. Weinstock C., Goodenough J.: Towards an Assurance Case Practice for Medical Devices, Software Engineering Institute, Technical Note CMU/SEI-2009-TN-018 (2009)
13. Ray A., Cleaveland R.: Constructing safety assurance cases for medical devices, in Proceedings of the 1st International Workshop on Assurance Cases for Software-Intensive Systems, pp. 40-45, IEEE Press (2013)
14. Wassyng A., Singh N.K., Geven M., Proscia N., Wang H., Lawford M., Maibaum T., Can product specific assurance case templates be used as medical device standards?, IEEE Design & Test, Issue 5 (2015)
15. Generic Infusion Pump Research Project website, https://rtg.cis.upenn.edu/gip/
16. Jee E., Lee I., Sokolsky O.: Assurance Cases in Model-Driven Development of the Pacemaker Software, Lecture Notes in Computer Science: Leveraging Methods of Formal Methods, Verification, and Validation 6416, pp. 343-356 (2010)
17. Ayoub A., Kim B., Lee I., Sokolsky O.: A Safety Case Pattern for Model-Based Development Approach, NASA Formal Methods LNCS vol. 7226, pp. 141-146 (2012)
18. Feng L., King A., Chen S., Ayoub A., Park J., Bezzo N., Sokolsky O., Lee I.: A Safety Argument Strategy for PC A Closed-Loop Systems: A Preliminary Proposal, 5th Workshop on Medical Cyber-Physical Systems 36, pp. 94-99 (2014)
19. Larson B.R.: Open PCA Pump Assurance Case, SAnToS research group, Kansas State University, http://openpcapump.santoslab.org/ (2014)
20. Kelly T., McDermid J.: Safety case construction and reuse using patterns, in Proceedings of SAFECOMP' 97, pp. 55-69 (1997)
21. Kelly T., McDermid J.: Safety case patterns – reusing successful arguments, In Proc. of IEE Colloquium on Understanding Patterns and Their Application to System Engineering, London, UK (1998)
22. Kelly T.: Arguing safety – a systematic approach to safety case management, PhD thesis, Department of Computer Science, University of York (1998)

23. Hawkins R., Kelly T., A Systematic Approach for Developing Software Safety Arguments, In proceedings of the 27th System Safety Society (SSS) International System Safety Conference (ISSC), 3-7 August 2009, Huntsville AL, USA (2009)
24. 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), Springer LNCS (2011)
25. Weaver R.: The Safety of Software – Constructing and Assuring Arguments, PhD Thesis, Department of Computer Science, University of York (2003)
26. Ye F.: Justifying the Use of COTS Components within Safety Critical Applications, PhD Thesis, Department of Computer Science, University of York (2005)
27. Alexander R., Kelly T., Kurd Z., McDermid J.: Safety cases for advanced control software: Safety case patterns, Technical Report, University of York (2007)
28. Hawkins R., Kelly T.: A software safety argument pattern catalogue, Technical report, University of York (2013)
29. Denney E., Pai G.: Safety Case Patterns: Theory and Applications, NASA/TM–2015–218492 Technical Report (2015)
30. Assurance Case patterns on-line catalogue, Gdańsk University of Technology, http://www.nor-sta.eu/en/en/news/assurance_case_pattern_catalogue
31. Hauge A., Stølen K.: A pattern-based method for safe control systems exemplified within nuclear power production, Safecomp 2012, LNCS vol. 7612, pp. 13–24 (2012)
32. Khalil M., Schätz B., Voss S.: A Pattern-based Approach towards Modular Safety Analysis and Argumentation, Embedded Real Time Software and Systems Conference (ERTS2014), Toulouse, France (2014)
33. Denney E., Pai G., Pohl J.: AdvoCATE: An Assurance Case Automation Toolset, in: SAFECOMP 2012 Workshops, LNCS Vol. 7613, pp. 8-21 (2012)
34. Denney E., Pai G.: A lightweight methodology for safety case assembly, In Proc. of 31st International Conference on Computer Safety, Reliability, and Security (SAFECOMP 2012), pp. 1-12, Springer Berlin Heidelberg (2012)
35. Hawkins R., Habli I., Kolovos D., Paige R., Kelly T.: Weaving an Assurance Case from Design: A Model-Based Approach, 2015 IEEE 16th International Symposium on High Assurance Systems Engineering (HASE) (2015)
36. Argevide: NOR-STA Argument Notation White Paper, https://www.argevide.com/sites/default/files/docs/Argevide%20WP2%20-%20NOR-STA%20argument%20notation.pdf
37. Jarzębowicz A., Wardziński A.: Integrating Confidence and Assurance Arguments, In: 10th IET System Safety and Cyber Security Conference, Bristol, UK (2015)
38. 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)